Indian Journal of Science and Technology, Vol 10(29), DOI: 10.17485/ijst/2017/v10i29/115882, August 2017
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Project Effort Estimation using COCOMO-2 Metrics with Fuzzy Logic

Rahul Kumar Yadav and S. Niranjan

Department of Computer Science, Mewar University, Gangrar, Chittorgarh – 312901, Rajasthan, India; niranjan.hug41@ gmail.com, er13rahulyadav@ gmail.com

Abstract

Objectives: Enhancement of the effort estimation by using Fuzzy Logic with COCOMO-II Effort Multiplier and comparing BRE, RE, VAF. **Methods/Statistical Analysis:** For the cost estimation, fuzzy logic is used with three membership functions such as Triangular, Trepezoidal and Bell. **Findings:** The results show that all three membership functions vary by 1-2 % in the effort estimation where as if compared; COCOMO 2 Effort varies by 5-10 % from Fuzzy output. The traditional method consumes a lot of time and a lot of methods are non mathematical due to which the predicted results may be irrelevant. **Application/Improvements:** The work has been tested on 5 projects. When the parameters for triangular function are executed, values for embedded 1, semidetached 2 are enhanced as compare to organic, semi detached 1, embedded 2. After the evaluation of trapezoidal function, values of semidetached 1 and embedded 1 are more as compare to others. Same as before, the values for embedded 1 and semidetached 2 are more in Bell membership function. Slight improvement has been seen in organic value. Similarly, for COCOMO II the results have been obtained.

Keywords: COCOMO-II, Cost Metrics, Fuzzy Logic, Software Effort Estimation

1. Introduction

Software effort estimation refers to measuring total effort in order to make a project successful. The success of a project may depend upon the following aspects;[1,2]

i. The total time spent in creating the project.
ii. The total human resources applied in order to complete the job.
iii. The quantity of the output.

There are various traditional methods of assessments of software effort. Some of the popular effort estimation techniques are:

i. Delphi technologies
ii. Three point estimation
iii. COCOMO-I and COCOMO-II Estimation[3–5]

The Delphi technique was based on surveys and information collected from the participants. The analysis is done as relevant, semi-relevant and non-relevant basis.

The three point technique is sort of work break down structure and in this, the entire project is broken in different segments for the analysis.[6] There are three types of estimation which can be performed:

i. Optimistic estimation
ii. Most likely estimation
iii. Pessimistic estimation

1.1 The COCOMO Model

COCOMO stands for Constructive Cost Model and it was developed.[7,8] It is used for the effort prediction based

*Author for correspondence

on some mathematical calculations and it can be applied to three categories of projects.

i. Organic: Project completed by small and organized team.
ii. Semi-detached: Project accomplished by group member who have a mixed experience.
iii. Embedded: Projects done with hard time schedule and constraints.

A lot of research workers have been used COCOMO as their primary tool to evaluate the effort of the projects. As per COCOMO, the effort is calculated by following formula:

$$\text{Effort} = a_1 * (\text{KLOC})^{a_2} \text{PM} \quad (1)$$

Where (KLOC) is the estimated size [generally lines of code (LOC)]

a_1 & a_2 are the constant and PM is the total months applied.

1.2 Fuzzy Logic

Fuzzy Logic (FL) has been developed and the term fuzzy logic is introduced.[9] The concept of FL was started with the concept of 'Fuzzy Set Theory' which is a class theory having un-sharp theories and taken as a classical set theory extension.[10]

An element x has membership of classical set B, as the universe X subset and is defined below:

$$\mu_B(X) = 1 \text{ if } x \epsilon B \quad (2)$$

$$\mu_B(X) = 0 \text{ if } x \epsilon \beta \quad (3)$$

The concept based on fuzzy logic has a direct connection with fuzzy concepts, namely, linguistic variables, fuzzy sets etc. Fuzzy logic could be divided into three types, namely,

i. Pure fuzzy logic,
ii. Takagi and Sugeno Fuzzy System
iii. Fuzzy logic System with Fuzzifier and Defuzzifier

Following are the steps being used to apply Fuzzy logic as a model are:[11]

Step 1: Fuzzification

It can change the crisp I/P (Input) to fuzzy set.

Step 2:

a) Fuzzy Rule Based System

The system of fuzzy logic utilizes Fuzzy IF-THEN rules.

b) Fuzzy Inference Engine

When the crisp I/P values are fuzzified in linguistic value, the inference engine executes the fuzzy rule base for deriving the linguistic values for the subsequent and the O/P (Output) linguistic variables.[12]

Step 3: Defuzzification

It transforms the fuzzy O/P into crisp O/P.

The paper is organized in five sections. The first section is the introduction part. Second section describes the problem statement. The third section is the proposed methodology followed by the fourth section which is the result evaluation. Fifth section is to conclude the entire paper with the directions for the future research workers.

2. Problem Statement

Software effort estimation is one of the key aspects to decide the success or failure of a project.[13] A lot of research workers have used COCOMO as their primary evaluator but there are following constraints with COCOMO-I.[14]

i. Less mathematical representation for idol real data relationship.
ii. Less number of parameters.

Due to this lag, COCOMO-II started getting adopted by the scientists and research workers. COCOMO-II has more number of evaluation parameters which works mathematically.

The problem of this research work is to enhance the effort estimation through COCOMO-II by adding Fuzzy Logic to it.

3. Proposed Methodology

As described in Section II, the proposed methodology would be enhancing the estimation technique by using Fuzzy logic to it.

3.1 COCOMO-II

COCOMO-II is a mathematical and idolized solution to evaluate the effort on a project. The following mathematical parameters have been utilized:

i. Required Software Reliability (RELY)

It's an evaluation of the threshold of the extent up to which software must perform.[15]

ii. Data Base Size Data

It is a measure of data requirement for the proceeding.[16]

iii. Product Complexity (CPLX)

The product complexity is divided into five sections:[17]

a) Control Operations CPLX
b) Computation Operation CPLX
c) Device Dependent CPLX
d) Data Management CPLX
e) User Interface CPLX

iv. Required Reusability (RUSE)

This is the required extra effort to complete the project.[18]

v. Execution Time (TIME)

Total execution time of the project is calculated by the execution time.[19]

vi. Storage Constraints (STOR)

Total amount of storage required for the project can be termed as shown in Table 1.[20]

vi. Programmer Capability (PCAP)

It is a parameter which is affected by the programmer's capability. It is influenced by the way a programmer's capability. It is influenced by the way a programmer codes.[21]

vii. Language and Tool Experience (LTEX)

It is also dependent upon the developer's tool experience and is shown in Table 2.[22]

viii. Multisite Development (SITE)

Ability to work on project from different places is called SITE.[23]

Table 1. Total Storage Amount

	Very Low	Low	Nominal	High	Very High	Extra High
STOR	N/A	N/A	50 % of storage	70 %	85 %	

Table 2. LTEX development tool

	Very Low	Low	Nominal	High	Very High
LTEX	2 months	6 months	1 year	3 years	6 +

Algorithm 1: Effort COCOMO-II (Evaluate COCOMO-II Effort (Cost Drivers)

// This algorithm evaluates the effort using COCOMO-II.
// The input to this algorithm is the cost drivers used in the project.
cost_drivers=Input .Data [Project_Data] // Reading the cost drivers
effort_drivers_COCOMO-II=1 // Initializing the effort multiplier to 0.
Effort_multiplier_COCOMO-II=

$$\sum_{i=1}^{n} cost_{drivers} * cost_{drivers}\,(i)$$

A=2.45; // Initializing Constants
B=1.01;
Size =Cost_driver (size); // Fetching size from cost driver

$$E = B + .01 * dataX \tag{4}$$

$$COCOMO - IIeffort = A * (size)E * Effort_{multiplier}; \tag{5}$$

Effort COCOMO-II = COCOMO-II effort
End function

3.2 Data Set

The dataset used to evaluate this project has been from promise.site.vottawa.ca/SE$_{repository}$/dataset/cocomonasa_2.arff. This data set contains records of 14 different years, starting from 1971 and ending at 1987.

The dataset evaluates the effort of 152 hours.

3.3 The Fuzzy Application

Fuzzy Logic is a rule based architecture which runs on binary pattern. It has Input set, associated with rule-sets based on the membership function. There are three membership functions which has been utilized in the proposed architecture.

3.3.1 The Triangular Membership Function

It has one maxima and two minima points as shown in Figure 1.[24]

There are two minima points and the shape is also triangular.[25]

3.3.2 The Trapezoidal Membership Function

The trapezoidal membership function attains the following shape as depicted in Figure 2:[26,27]

3.3.3 Bell Membership Function

Bell Membership function is similar to that of Gaussian Membership Function as shown in Figure 3.[28,29]

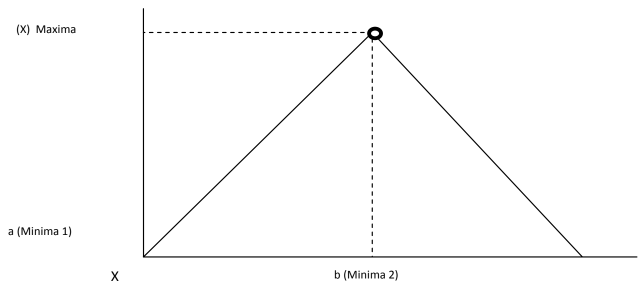


Figure 1. Triangular Membership function.

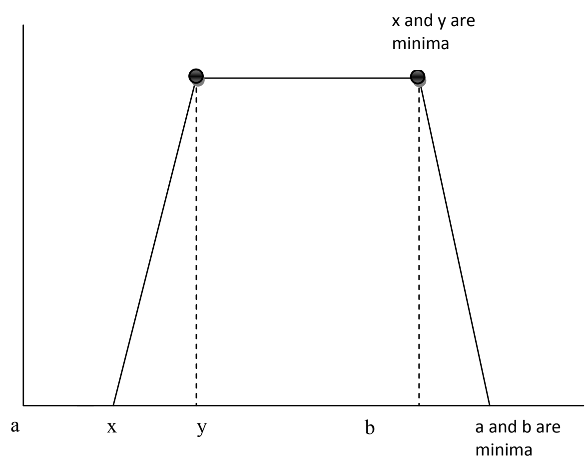


Figure 2. Trapezoidal Membership function.

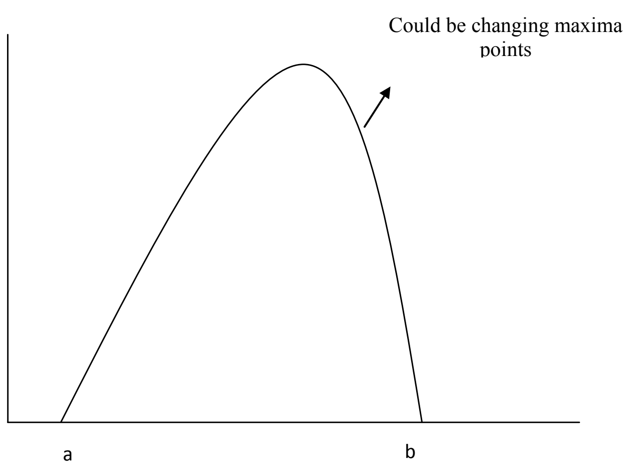


Figure 3. Bell Membership function.

Algorithm 2: Function generate rule sets (type, time, cpixy, PCAP, RUSE, site, rely, stor)

// This algorithm generates the ruleset for the type of membership function provided to it.

// The input set takes the types of membership function.

If (type==1)

// Mean triangular membership function will be selected.

i. Effort would be least

If (((cplxy<0.33) && timey<1) && (rusey > 0.66 && pcapy > 0.66&& sitey >0) &&(rely<0. 33 && story<=0.33))

ii. Effort would be moderate

If (((cplxy<0.50) && cplxy>0.33) && (timey < 1.5 && timey > 1))&&(rusey>0.66 &&pcapy 0.66 && sitey>0) && ((relyy<0. 50&&rely>0.33) && (story <=0.50 && story>0.33))

iii. Effort would be maximum

If (((cplxy<0.0) && timey<1.5) && (rusey > 0.66 && pcapy > 0.66&& sitey >0) &&(rely<0. 0.60 && story>=0.60))

If (type==2)

i. Effort would be least

If (((cplxy<0.33) && timey<1) && (rusey > 0.66 || pcapy > 0.66 && sitey >0) &&(rely<0. 33 && story<=0.33))

ii. Effort would be moderate

If (((cplxy<0.50) && cplxy>0.33) && (timey <1.5|| timey > 1)) && (rusey > 0.66) && (pcapy>0.66 && sitey>0) && ((relyy<.50 && relyy>0.33) && (story<=0.50&& story>0.33))

iii. Effort would be maximum

if (((cplxy<.0) &&timey<1.5) && (rusey>0.66 || pcapy> 0.66 && sitey>0) && (relyy>0.60 && story>=.060))

Else

iv. Effort would be least

if (((cplxy<.40)&&timey<1.2) &&(rusey==0.66 || pcapy==0.66 &&sitey>0)&&(relyy<0.33 && story<=0.33))

v. Effort would be moderate

if (((cplxy<.50 && cplxy>.40)&&(timey==1.5)) &&(rusey>0.76 &&pcapy>0.66 &&sitey>0)&&((relyy<.50 && relyy>.33) && (story<=0.50&& story>0.33))

vi. Effort would be maximum

If (((cplxy<.0) &&timey<1.5) &&(rusey>0.89 || pcapy>0.66 &&sitey>0)&&(relyy>0.60 && story>=0.60))

Based on the rule sets, the effort through fuzzy logic has been calculated.

The following equations have been utilized to calculate the effort multi

Effort Multiplier= $\sum_{i=1}^{n} svm * data(Rely)$

Effort= $A * ((size \uparrow E) * effort_{\downarrow}Multiplier$

$$E = B + 0.01 * scalefactor$$

$$A = 2.45$$
$$B = 1.01$$

A part from the Effort, the following parameters has been also evaluated.

i. Variance Accounted For (VAF)

$$\% \text{ VAF} = \left[1 - \frac{\text{var}(\textit{Measured Effort} - \textit{Estimated Effort})}{\text{var}(\textit{Measured Effort})}\right] \times 100$$

ii. Mean Absolute Relative Error (MARE)

$$\% \text{MARE} = \text{mean}\left[\frac{abs(\textit{Measured Effort} - \textit{Estimated Effort})}{(\textit{Measured Effort})}\right] \times 100$$

iii. Variance Absolute Relative Error (VARE)

$$\% \text{ VARE} = \text{var}\left[\frac{abs(\textit{Measured Effort} - \textit{Estimated Effort})}{(\textit{Measured Effort})}\right] \times 100$$

iv. Prediction (n)

Prediction at level n is defined as the % of projects that have absolute relative error less than n.

v. Balance Relative Error (BRE)

$$\text{BRE} = \frac{|E - \hat{E}|}{\min(E, \hat{E})}$$

Where E = Estimated Effort, $\hat{E}$ =Actual Effort.

$$\text{Absolute Relative Error (RE)} = \frac{|E - \hat{E}|}{E}$$

A model which gives higher VAF is better than that which gives lower VAF. A model which gives higher Pred(n) is better than that which gives lower Pred(n). A model which gives lower MARE is better than that which gives higher MARE. A model which gives lower VARE is better than that which gives higher VARE. A model which gives lower BRE is better than that which gives higher BRE.

4. Results and Simulation

The following simulation results have been evaluated.

As discussed in the previous sections of the paper, three types of membership functions, triangular, trap-

Table 3. Parameters for Triangular function

	VAF	MARE	VARE	RE	BRE	AE	BRE
Semidetached 1	42.4674	57.5326	57.5326	0.36521	0.57533	25.2	39.698
Organic	50.0024	49.9976	49.9976	0.9999	0.9999	150	75.003
Semidetached 2	63.3888	36.6112	36.6112	0.268	0.36611	432	590.16
Embedded 1	63.1024	36.8976	36.8976	0.26953	0.369898	409	559.912
Embedded 2	74.0769	25.9231	25.9231	0.20586	0.25923	38	47.85

Table 4. Parameters for Trapezoidal function

	VAF	MARE	VARE	RE	BRE	AE	BRE
Semidetached 1	42.4678	57.5352	57.5352	0.36522	0.57535	25.2	39.698
Organic	50.00.33	49.9967	49.9967	0.99987	0.99987	150	75.004
Semidetached 2	63.3865	36.6135	36.6135	0.26801	0.36614	432	590.17
Embedded 1	63.1001	36.8999	36.8999	0.26954	0.369	409	559.92
Embedded 2	74.0748	25.9252	25.9252	0.20588	0.25925	38	47.851

Table 5. Parameters for Bellfunction

	VAF	MARE	VARE	RE	BRE	AE	BRE
Semidetached 1	42.4952	57.5048	57.5048	0.3651	0.57505	25.2	39.691
Organic	49.9936	50.0064	50.0064	1.003	1.003	150	74.99
Semidetached 2	63.4129	36.5871	36.5871	0.26787	0.36587	432	590.056
Embedded 1	63.1266	36.8734	36.8734	0.2694	0.36873	409	559.812
Embedded 2	74.0991	25.9009	25.9009	0.20572	0.25901	38	47.842

ezoidal and bell membership function has been used. Results have been evaluated using the same.

The evaluation has been done on the basis of 5 tested projects. More projects could have been also evaluated but it would have been a little difficult for one paper to represent all the results for more than 10 projects. Table 3 and Figure 4 represent the comparison of the parameters for the triangular function.

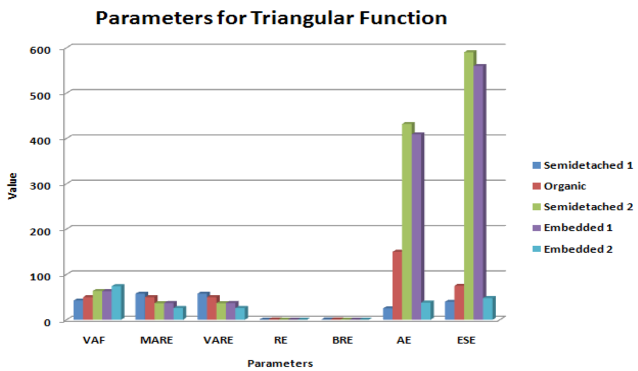


Figure 4. Parameters for Triangular function.

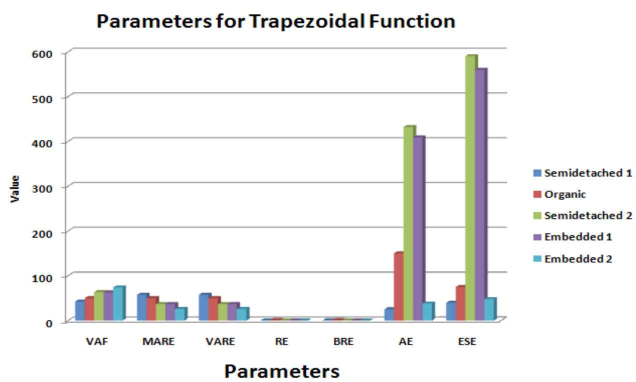


Figure 5. Parameters for Trapezoidal function.

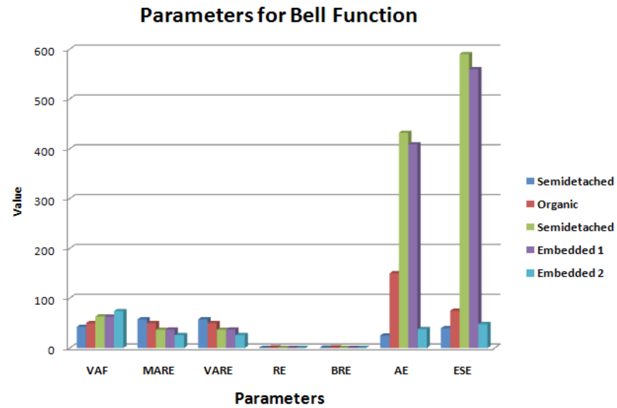


Figure 6. Parameters of Bell membership function.

The same parameters have been used for the evaluation of trapezoidal function also. Table 4 and Figure 5 represent the comparison of the parameters for the trapezoidal function. The only difference is that, it uses trapezoidal membership function.

The same parameters have been used for the evaluation of bell function also. Table 5 and Figure 6 represent the comparison of the parameters for the trapezoidal function. The only difference is that, it uses bell membership function instead of trapezoidal membership function.

Table 6 and Figure 7 represent the comparison of the parameters of COCOMO II. In the above figure there are five different types of projects are used and we calculate and compare the parameters among of all types. The total effort has also been calculated for Fuzzy and Cocomo 2. The following figure represents the effort comparison for COCOMO 2 and FUZZY model.

Figure 8 represents the Effort plotted against different sets of data. The tabular results show the comparison between the COCOMO2 effort and the effort calculated by the Fuzzy models.

Table 6. Parameters of COCOMO II

	VAF	MARE	VARE	RE	BRE	AE	ESE
Semidetached 1	36.699	63.301	63.301	0.63301	0.38763	25.2	41.151
Organi	51.8334	48.1666	48.1666	0.92926	0.92926	150	77.75
Semidetached 2	58.3865	41.6135	41.6135	0.41614	0.29385	432	611.77
Embeded 1	58.0896	41.9104	41.9104	0.4191	0.29533	409	580.41
Embeded 2	69.466	30.534	30.534	0.30534	0.23392	38	49.6

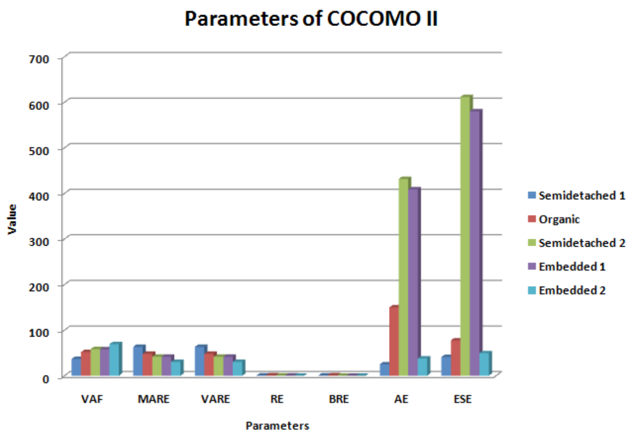
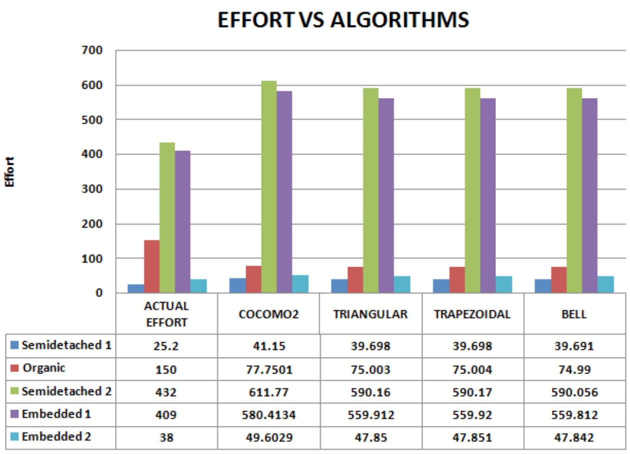


Figure 7. Parameters of COCOMO II.



	ACTUAL EFFORT	COCOMO2	TRIANGULAR	TRAPEZOIDAL	BELL
Semidetached 1	25.2	41.15	39.698	39.698	39.691
Organic	150	77.7501	75.003	75.004	74.99
Semidetached 2	432	611.77	590.16	590.17	590.056
Embedded 1	409	580.4134	559.912	559.92	559.812
Embedded 2	38	49.6029	47.85	47.851	47.842

Figure 8. Effort plotted against algorithms.

5. Conclusion

Software effort estimation is one of the most comprehensive and required operations in order to judge whether the project is going to succeed or not. The proposed algorithm has opted Fuzzy logic for the cost estimation and three membership functions namely Triangular, Trapezoidal and Bell. A lot of parameters including Effort Multiplier, BRE, RE, VAF have also been computed to compare the results. 3 series of rule sets has been used to evaluate the Fuzzy outcome. The results show that all three membership functions vary by 1-2 % in the effort estimation where as if compared; COCOMO 2 Effort varies by 5-10 % from Fuzzy output. The current research work has opened a lot of aspects for the future research workers. The future research workers may add more rules to the fuzzy member ship function. Gaussian Membership function can also be tried. It would be also interesting to see how Artificial Intelligence affects the process. The dataset used in this procedure has been adapted from NASA. Other datasets can also be tried and tested.

6. References

1. Molokken K, Magen J. A review of software surveys on software effort estimation. Empirical Software Engineering. International Symposium on IEEE; 2003. p. 223. Crossref
2. Mukhopadhyay T, Vicinanza SS, Prietula JM. Examining the feasibility of a case-based reasoning model for software effort estimation. MIS quarterly. 1992; 16(2):155–71. Crossref
3. Schmidt R, Lyytinen K, Cule P, Keil M. Identifying software project risks: An international Delphi study. Journal of management information systems. 2001; 17(4):5–36. Crossref
4. Shepperd M, Schofield C. Estimating software project effort using analogies. IEEE Transactions on software engineering. 1997; 23(11):736–43. Crossref
5. Basha S, Dhavachelvan P. Analysis of empirical software effort estimation models; 2010. p. 1–10.
6. Attarzadeh I, Ow SH. Improving estimation accuracy of the COCOMO II using an adaptive fuzzy logic model. Fuzzy Systems (FUZZ). IEEE International Conference on IEEE; 2011. p. 2458–64. Crossref
7. Boehm BW, Madachy R, Steece B. Software cost estimation with Cocomo II with Cdrom. Prentice Hall PTR; 2000.
8. Boehm BW, Valerdi R. Achievements and challenges in Cocomo-based software resource estimation. IEEE software. 2008; 25(5):1–10. Crossref
9. Zadeh LA. Fuzzy logic neural networks and soft computing. Communications of the ACM. 1994; 37(3):77–85. Crossref
10. Idri A, Alain A, Kjiri L. COCOMO cost model using fuzzy logic. 7th International Conference on Fuzzy Theory & Techniques; 2000. p. 1–4.
11. Mittal A, Parkash K, Mittal H. Software cost estimation using fuzzy logic. ACM SIGSOFT Software Engineering Notes. 2010; 35(1):1–7. Crossref
12. Xu Z, Khoshgoftaar TM. Identification of fuzzy models of software cost estimation. Fuzzy Sets and Systems. 2004; 145(1):141–63. Crossref
13. Clark B, Chulani SD, Boehm B. Calibrating the COCOMO II post-architecture model. Software Engineering. Proceedings of International Conference on IEEE; 1998. p. 1–10. Crossref
14. Menzies T. Selecting best practices for effort estimation. IEEE Transactions on Software Engineering. 2006; 32(11):1–132. Crossref
15. Huang LG, Boehm B. How much software quality investment is enough: A value-based approach? IEEE software. 2006; 23(5):88–95. Crossref
16. Chen Z. Finding the right data for software cost modeling. IEEE Software. 2005; 22(6):38–46. Crossref

17. Huang X. Improving the COCOMO model using a neuro-fuzzy approach. Applied Soft Computing. 2007; 7(1):29–40. Crossref
18. Briand LC, Emam EK, Surmann D. Wieczorek I, Maxwell KD. An assessment and comparison of common software cost estimation modeling techniques. Software Engineering Proceedings of International Conference; 1999. p. 313–32. Crossref
19. Boehm B. Cost estimation with COCOMO II. Edition Upper Saddle River. NJ: Prentice-Hall; 2000.
20. Sharma TN. Analysis of software cost estimation using COCOMO II. International Journal of Scientific & Engineering Research. 2011; 2(6):1–5.
21. Chulani S, Boehm B, Steece B. Bayesian analysis of empirical software engineering cost models. IEEE Transactions on Software Engineering. 1994; 25(4):573–83. Crossref
22. Baik J, Boehm B, Steece BM. Disaggregating and calibrating the CASE tool variable in COCOMO II. IEEE Transactions on Software Engineering. 2002; 28(11):1009–22. Crossref
23. Chulani S, Boehm B. Modeling software defect introduction and removal: COQUALMO (COnstructive QUALity MOdel). Technical Report USC-CSE-99-510, University of Southern California; Center for Software Engineering; 1999.
24. Castro JL. Fuzzy logic controllers are universal approximators. IEEE transactions on Systems Man and Cybernetics. 1995; 25(4):629–35. Crossref
25. Zhao J, Bose BK. Evaluation of membership functions for fuzzy logic controlled induction motor drive. IEEE 2002 28th Annual Conference. 2002; 1:229–34.
26. Uddin MN, Radwan TS, Rahman MA. Performances of fuzzy-logic-based indirect vector control for induction motor drive. IEEE Transactions on Industry Applications. 2002; 38(5):1219–25. Crossref
27. Bowles JB, Peláez CE. Fuzzy logic prioritization of failures in a system failure mode effects and criticality analysis. Reliability Engineering & System Safety. 1995; 50(2):203–13. Crossref
28. Liang Q, Mendel JM. Interval type-2 fuzzy logic systems: Theory and design. IEEE Transactions on Fuzzy systems. 2000; 8(5):535–50. Crossref
29. Rao MVC, Prahlad V. A tunable fuzzy logic controller for vehicle-active suspension systems. Fuzzy sets and systems. 1997; 85(1):11–21. Crossref