

# Comparative Analysis of COCOMO81 using Various Fuzzy Membership Functions

Pooja Jha and K.S.Patnaik

Department of Computer Science & Engineering, BIT Mesra, Ranchi (Jharkhand)

## ABSTRACT

Software Estimation has always been one of the prompting challenges for the software engineers. Software cost estimation techniques helps in forecasting the amount of effort required to develop software. Constructive Cost Model (COCOMO) is considered to be the most widely used model for effort estimation. Cost drivers have great influence on the COCOMO and this paper investigates the role of cost drivers in improving the precision of effort estimation using different membership functions. Fuzzy logic-based estimation models are more suitable when formless and inaccurate information is to be used. The proposed fuzzy COCOMO model consists of a collection of linear sub-models joined together smoothly using fuzzy membership functions. This paper focus on the comparative analysis of COCOMO81 using various fuzzy membership functions. The present work is based on COCOMO81 dataset and the experimental part of the study illustrates the approach and compares it with the standard version of the COCOMO81. It has been found that Fuzzy based COCOMO model gives better performance when compared to the COCOMO81, demonstrating a smoother transition in its intervals, and the achieved results were closer to the actual effort.

## Keywords:

Software cost estimation, COCOMO81, EAF, Fuzzy logic, Membership Functions

## 1. INTRODUCTION

Managing software projects revolves around software cost estimation as it is the most critical task. It is observed that there exists an avoidable split between the estimated costs and the actual costs of the software projects [1]. Hence, accurate cost estimates are highly desired during the early phases of development. The precision of the effort estimate is very important for software industry because both overestimates and underestimates of the software effort are harmful to software companies. Most commonly used techniques for software cost estimation are the algorithmic models such as COCOMO [2][3][4], IBM-FSD [5], PUTNAM-SLIM [6], SPQR [7] and function points analysis [8][9]. Constructive Cost Model (COCOMO) is proposed to be extended by incorporating the concept of fuzziness into the measurements of cost drivers; fuzzy set theory is used rather than classical intervals to represent the linguistic values [2]. Incorporating fuzziness in cost estimation is more natural and mimics the way in which humans interpret linguistic values. The conventional quantization in COCOMO is substituted by using various membership functions at fuzzy interval values have been in use [10]. It has been found that Gaussian Bell function is performing better than Gaussian, trapezoidal or triangular fuzzy membership functions, as it demonstrates a smoother transition in its intervals, and the achieved results were

closer to the actual effort.

This paper is organized as follows: Section 2 provides the general idea of the methods employed in this paper. Section 3 briefly discusses the statement of the problem Section 4 describes the related work done for estimating the effort through different fuzzy logic approaches. Section 5 presents experimental results and discussion used in the application of various functions to COCOMO81 using fuzzy logic tool box. Section 6 concludes that the accuracy of effort estimation can be improved through the proposed model and the estimated effort can be very close to the actual effort.

## 2. LITERATURE REVIEW

### 2.1 ALGORITHMIC MODELS

Forecasting effort using algorithmic models depends upon the accurate estimate of either size of software in terms of lines of code (LOC), number of user screens, interfaces, complexity, etc. which is quiet difficult when uncertainty prevails in the project [11]. Boehm was the first researcher to consider software engineering economically. He came up with a cost estimation model, COCOMO-81 in 1981, after investigating a large set of data from TRW in the 1970s [12]. COCOMO assumes that the effort grows more than linearly on software size i.e.

$$\text{months} = a * \text{KDSI}^b * c$$
Here, 'a' and 'b' are domain-specific parameters; 'KDSI' is estimated directly or computed from a function point analysis; and 'c' is the product of over 15 effort multipliers (EMs). Putnam also developed an early model known as SLIM in 1978[13]. COCOMO and SLIM both make use of data from past projects and are based on linear regression techniques. Both COCOMO and SLIM take number of lines of code (about which not much is known during early in the project) as the major input to their models. A survey on these algorithmic models and other cost estimation approaches is presented by Boehm et. al.[14]. Algorithmic models such as COCOMO are unable to present suitable solutions that take into consideration technological advancements [15]. This is because; these models are often unable to capture the complex set of relationships (e.g. the effect of each variable in a model to the overall prediction made using the model) that are evident in many software development environments [16]. They are not flexible enough to adapt into new environment, cannot handle categorical data as well as they lack reasoning capabilities. These drawbacks led to discover ways for exploring non-algorithmic models.

### 2.2 SOFT-COMPUTING BASED NON-ALGORITHMIC MODELS

Soft computing is a conglomerate of methodologies centering in fuzzy logic (FL), artificial neural networks (ANN) and evolutionary computation (EC). This section

discusses some of the non-algorithmic models for software development effort estimation. Soft computing encompasses methodologies centering in fuzzy logic (FL), artificial neural networks (ANN) and evolutionary computation (EC). These methodologies handle real life vague situations by providing flexible information processing capabilities. The first realization of the fuzziness of several aspects of one of the best known [17], most successful and widely used model for cost estimation, COCOMO, was that of Fei and Liu [12]. They observed that an accurate estimate of delivered source instruction (KDSI) cannot be made before starting the project; therefore, it is unreasonable to assign a determinate number for it. Jack Ryder investigated the application of fuzzy modeling techniques in COCOMO and the Function-Points models, respectively [11]. Fuzzy Logic was applied to the cost drivers of intermediate COCOMO model (the most widely used version) due to its relatively high estimation accuracy than the basic version, which is quite comparable to the detailed version [18]. The key project attribute "size" to estimate the software development effort was not considered. Musilek et al. applied fuzzy logic to represent the mode and size as input to COCOMO model by presenting a two-stage implementation called simple F-COCOMO model and augmented F-COCOMO model, respectively [19]. Ahmed et al. fuzzified two parts of COCOMO model as nominal effort estimation and the adjustment factor and proposed a fuzzy logic framework for effort prediction by integrating the fuzzified nominal effort and the fuzzified effort multipliers of the intermediate COCOMO model [11]. Boetticher has described a neural network approach for characterizing programming effort based on internal product measures [20]. A study shows that genetic programming can offer significant improvements in software estimation accuracy but this depends on the measure and interpretation of accuracy used [21]. The previous research reveals that all of the soft computing-based software effort prediction models that exist, lack in some aspect or the other. Uncertainty regarding suitability of prediction technique to particular type of prediction problem is evident [22]. A compelling demand to develop a single soft computing based model which handles tolerance of imprecision in the input at the preliminary phases of software engineering, addresses the fuzzification of one of the key attribute i.e. size of the project, incorporates expert's knowledge in a well-defined manner, allows total transparency in the prediction system by prediction of results through rules or other means, adaptability towards continually changing development technologies and environments always exists [22].

### 2.3 FUZZY LOGIC BASED FRAMEWORK

The paper aims to develop a fuzzy logic based framework to deal with the imprecision and uncertainty in the data during the early stage of software development, thus predicting the effort more accurately. The framework is based on existing COCOMO model. COCOMO81 is said to be the best known, most plausible, and most cited of all traditional models [15] [17] [19]. The COCOMO model is a set of three models: basic, intermediate, and detailed [2]. The various inputs to the COCOMO model are the (a) estimated size in the software in Kilo Source Lines of Code (KSLOC), (b) development mode, as constant B (scaling factor) and (c) 15 cost drivers [2] [23]. The development mode depends on one of the three categories of software development modes: organic, semi-detached, and embedded; reflecting the

difficulty level of development. It takes only three values, {1.05, 1.12, 1.20}. Cost drivers are the adjusting factors that influence the effort. Cost drivers have up to six levels of rating: Very Low, Low, Nominal, High, Very High, and Extra High. Each rating has a corresponding real number (effort multiplier), based upon the factor and the degree to which the factor can influence productivity. A membership function can be triangular, trapezoidal, gaussian, bell etc. The proposed framework will allow fuzzy and expert knowledge incorporation into the system.

### 3. PROBLEM STATEMENT

Inaccuracy in software cost estimation affects many software projects and this is apparent for decades. Poor estimates led projects to exceed budget and schedule, as well as, to terminate entirely in many cases. The ability to accurately estimate software development time, cost, and manpower, replaces older methodologies with the newer methodologies. Therefore, an accurate software cost estimation model is highly desirable in software project management.

### 4. RESEARCH METHOD

This section firstly introduces the characteristics and strengths of COCOMO and Fuzzy Logic. Then, a FL-COCOMO is explained in brief using different fuzzy membership functions.

#### 4.1 COCOMO

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current project characteristics. COCOMO consists of a hierarchy of three levels. The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes (Cost Drivers). Intermediate COCOMO takes these Cost Drivers into account and Detailed COCOMO additionally accounts for the influence of individual project phases. Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of source lines of code (SLOC) COCOMO applies to Organic, Semi-detached and Embedded classes of projects.

The basic COCOMO equations take the form

- Effort Applied (E) =  $a_b(KLOC)b_b$  [ man-months ]
- Development Time (D) =  $c_b(Effort Applied)^{d_b}$  [months]
- People required (P) = Effort Applied / Development Time [count]

where, KLOC is the estimated number of delivered lines (expressed in thousands) of code for project. The  $a_b$ ,  $b_b$ ,  $c_b$  and  $d_b$  are the coefficients. Intermediate COCOMO computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes. Each of the 15 attributes receives a rating on a six-point scale that ranges from "very low" to "extra high" (in importance or value). The product of all effort multipliers results in an effort adjustment factor (EAF). Detailed COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process.

## 4.2 Fuzzy Logic COCOMO (FL-COCOMO)

Using fuzzy sets in EMs and software size attributes can be specified by distribution of their possible values instead of using fixed values, which is represented in the form of a fuzzy set. Imprecision and uncertainty at the input level of the COCOMO81 yields uncertainty at the output level. Converting the software attributes to respective fuzzy set improves the accuracy of the software attributes which resulted in estimation accuracy. The software attributes in COCOMO is converted to fuzzy variables based on fuzzification process and terms Very Low (VL), Low (L), Nominal (NOM), High (H), Very High (VH) and Extra-High (XH) were defined for 15 cost drivers and one software size. At the first step, all software attributes are converted to corresponding fuzzy sets and variables. An FIS is developed that takes size and mode as fuzzy inputs to give the estimated effort in terms of size and mode. Individual FIS is modeled to obtain fuzzified EM for each of the cost drivers. Effort Adjustment Factor (EAF) is then calculated as the product of individual fuzzy EM for each of the cost drivers [24]. The equation is represented as in equation (1).

$$\text{Fuzzy}_{EMij} = \sum_{i=1}^n \mu_{A_i}^{v_i} * EM_{ij} \quad (1)$$

Finally, the Fuzzy effort is calculated as the product of fuzzy mode, size and fuzzy EM obtained [25], as shown in equation (2).

$$F\_Size\_EM_{ij} = \sum_{j=1}^{k_i} \mu_{A_j}^{v_j} * EM_{ij} * Size \quad (2)$$

where the  $\mu_{A_j}^{v_j}$  is the membership function of the fuzzy set  $A_j$  associated with the cost driver  $v_j$  and  $\mu_A$  is the membership function associated with the size.

The new FL-COCOMO is established based on COCOMO 81 and Fuzzy Logic. FL-COCOMO consists of 15 cost drivers and size as inputs to the model and Effort as output as shown in figure 1 below:

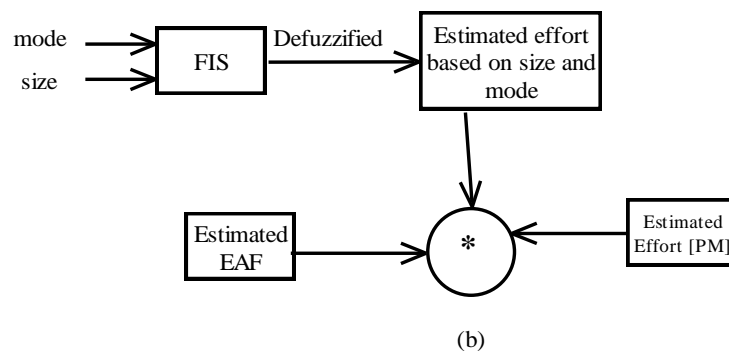
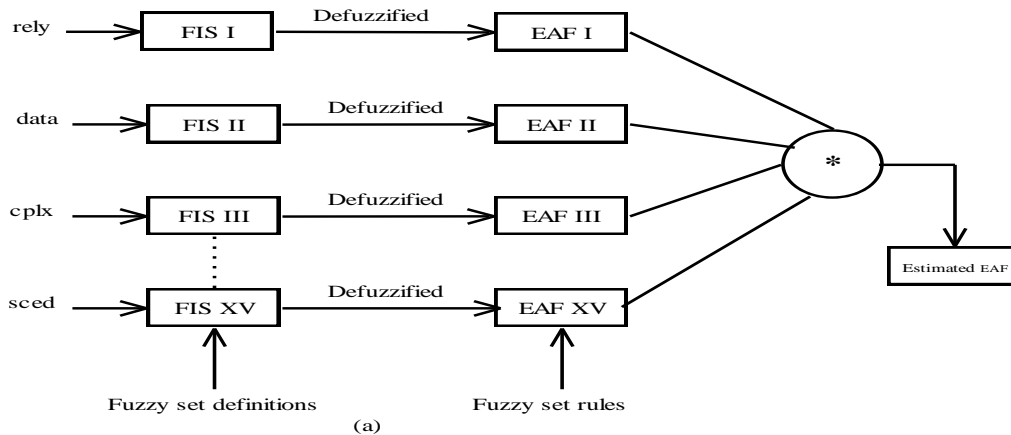


Fig 1: a) Fuzzification of cost drivers to obtain Estimated EAF b) Fuzzification of size and calibration of Estimated Effort

The fuzzy sets corresponding to the various associated linguistic values for each software attribute were defined using different membership functions. For example fuzzification of Analyst Capability (ACAP), Analyst experience (AEXP), Programmer Capability (PCAP), Language Experience (LEXP) and Virtual Experience (VEXP) description are given as in Table 1, 2, 3, 4 and 5 respectively.

Table 1:

Effort Multiplier : Analyst Capability ACAP					
Definition	VL	L	N	H	VH
Percentile	15	35	55	75	90
Value	1.46	1.19	1.00	0.86	0.71

Table 2:

Effort Multiplier : Analyst Experience AEXP					
Definition	VL	L	N	H	VH
Percentile	<=4 months	1 year	3 years	6 years	12 years
Value	1.29	1.13	1.00	0.91	0.82

Table 3:

Effort Multiplier : Programmer Capability PCAP					
Definition	VL	L	N	H	VH
Percentile	15	35	55	75	90
Value	1.42	1.17	1.00	0.86	0.70

Table 4:

Effort Multiplier : Language Experience LEXP				
Definition	VL	L	N	H
Percentile	<=1 month	4 months	1 year	3 years
Value	1.14	1.07	1.00	0.95

Table 5:

Effort Multiplier : Virtual Experience VEXP				
Definition	VL	L	N	H
Percentile	<=1 month	4 months	1 year	3 years
Value	1.21	1.10	1.00	0.90

The Fuzzy Inference System (FIS) is used in the fuzzification, fuzzy calculations, fuzzy rule generation, and defuzzification process of the FL-COCOMO. The FIS supports Mamdani and Sugeno fuzzy methods.

COCOMO used Mamdani FIS method due to its intuitive, widespread acceptance and well suited for human input nature. Figure 2 show the fuzzification of Personnel attributes EM using MATLAB.



Fig 2: Fuzzification of various Personnel attributes EM using FIS tool in the MATLAB software.

Fuzzy rules were defined for each of the cost drivers and size based on various modes. Almost 75 rules for the cost drivers were developed and separate rules for effort depending on size and modes were defined. Some of the rules are as:

- If SCED is low then EFFORT is low
- If DATA is high then EFFORT is high
- If ACAP is nominal then EFFORT is nominal
- If PCAP is very high then EFFORT is very low
- ...

This is represented in MATLAB as shown in figure 3 below:

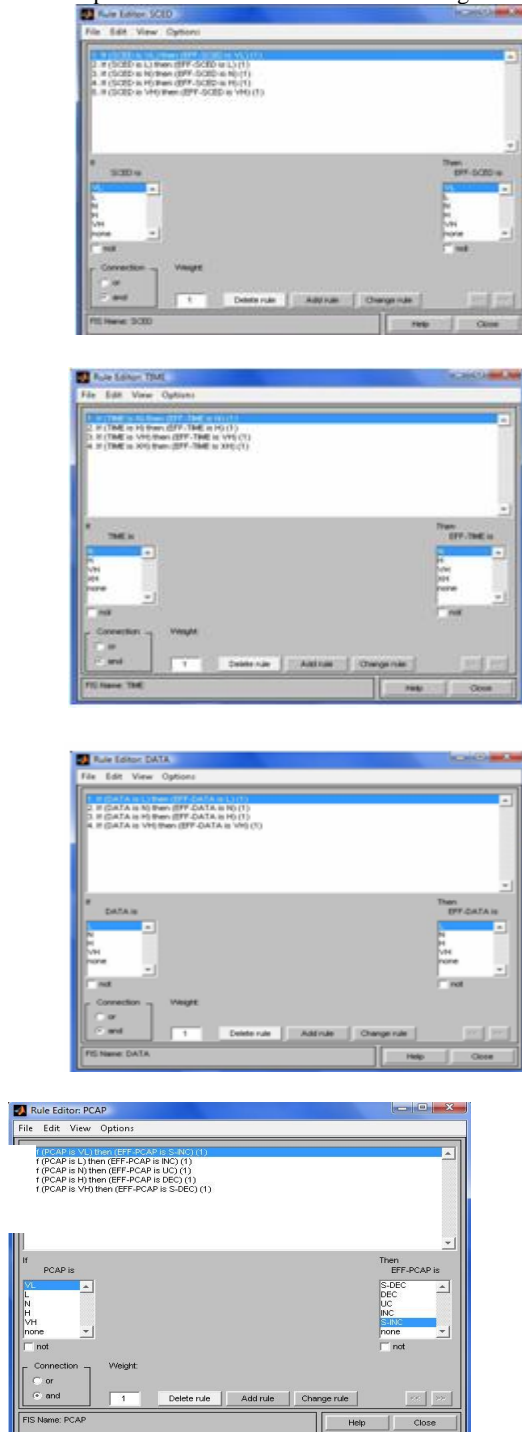


Fig 3: Rules generated for cost drivers

The last step is defuzzification of effort variable. The defuzzification of the output “Effort” is performed using the Center of Area (COA) or Centroid method.

## 5. RESULTS AND DISCUSSIONS

FL-COCOMO was evaluated using public domain COCOMO 81 data set, Data Set 1, which includes 63 historical project data. This data set is in public domain and available in <http://promisedata.org>.

### 5.1 Evaluation Method

The Mean Magnitude of Relative Error (MMRE) and the probability of a project having a relative error of less than or equal to L (PRED(L)) are the most commonly used evaluation criterion.

The Magnitude of Relative Error (MRE) is defined in equation (3) as follows:

$$MRE_i = \frac{|\text{Actual Effort}_i - \text{Predicted Effort}_i|}{\text{Actual Effort}_i} \quad (3)$$

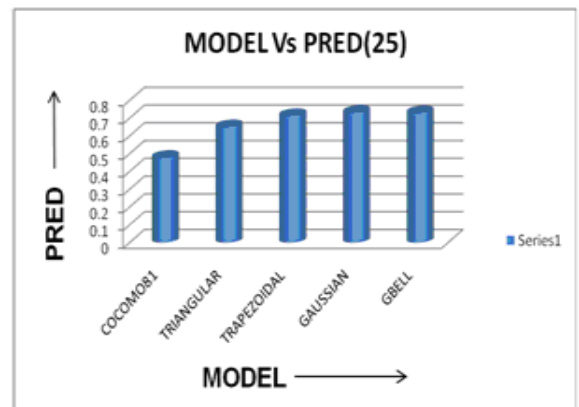
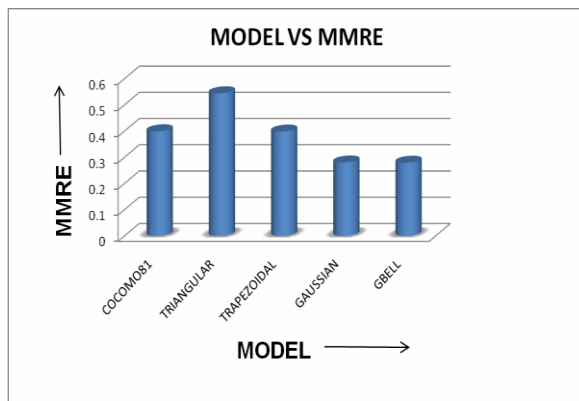
The MRE value was calculated for each observation  $i$  that effort is estimated at that observation. The aggregation of MRE over multiple observations ( $N$ ) can be achieved through the Mean MRE (MMRE) in equation (4) as follows:

$$MMRE = \frac{1}{N} \sum_i^N MRE_i \quad (4)$$

Prediction at level  $L$  is given by  $PRED(L) = k / N$ , where  $k$  is number of observation where MRE is less than or equal to  $L$ .  $N$  gives the number of observations. The paper observes for PRED(25). A comparison is made between COCOMO and FL-COCOMO using different membership functions and MMRE and PRED(25) is evaluated for better understanding. The comparison is shown in the Table 6 as below. Figure 4 gives the graphical comparison between COCOMO and FL-COCOMO using various membership functions.

**Table 6: Comparison between obtained results from COCOMO 81 and FL-COCOMO in terms of MMRE and PRED(25)**

MODEL		EVALUATION	
		MMRE	PRED (25)
	<b>COCOMO 81</b>	<b>0.399</b>	<b>0.479</b>
	<b>FL-COCOMO (Using Triangular function)</b>	<b>0.544</b>	<b>0.651</b>
	<b>FL-COCOMO (Using Trapezoidal function)</b>	<b>0.398</b>	<b>0.714</b>
	<b>FL-COCOMO (Using Gaussian function)</b>	<b>0.280</b>	<b>0.732</b>
	<b>FL-COCOMO (Using GBell function)</b>	<b>0.279</b>	<b>0.730</b>



**Figure 4: Comparison between COCOMO and FL-COCOMO with different membership functions in terms of MMRE and PRED(25)**

The percentage of improvement of FL-COCOMO is calculated based on the difference between the two models divided by the value of the COCOMO. Table 7, shows the

improvement in the model after applying the fuzzy membership functions.

**Table 7: Improvement using different fuzzy membership functions.**

Model	Evaluation		Improvement MMRE (%)	Improvement PRED (%)
	MMRE	PRED (25)		
COCOMO	0.399	0.479		
FL-COCOMO (Using Triangular function)	0.544	0.651	- 36.34	35.9
FL-COCOMO (Using Trapezoidal function)	0.398	0.714	0.25	49.0
FL-COCOMO (Using Gaussian function)	0.280	0.732	29.82	52.8
FL-COCOMO (Using GBell function)	0.279	0.730	30.0	52.4

In Table 7, the MMRE for FL-COCOMO using Triangular function shows that COCOMO is performing better than the FL applied. In terms of MMRE, the percentage of improvement is visible from FL-COCOMO, using Trapezoidal function and is about 0.25 %, whereas in FL-COCOMO, using Gaussian function is 29.82 %. The improvement in terms of MMRE is best for GBell of about 30 %. Pred (25 %) is shown best for Gaussian and GBell FL-functions being 52.8 % and 52.4 % respectively. It is shown that using FL can improve the estimation accuracy and it can be used as alternative to apply to the other software cost estimation models to improve estimation accuracy.

## 6. CONCLUSION

Accurate and reliable estimation of software time, cost, and manpower, especially in the early phase of software development is crucial issue in software project development. The objective of this paper was to examine the

application of applying fuzzy logic in software cost estimation for obtaining more accurate results. An adaptive software cost estimation model by incorporating fuzzy logic technique can handle imprecision and uncertainty of software attributes.

FL-COCOMO produced better estimation results than the COCOMO using evaluation criterion MMRE and PRED (25%). Furthermore, the percentage of improvement for the

FL-COCOMO varies from 0.25 % to 30 % and 35.9 to 52.8 % in terms of MMRE and PRED (25%).

## 7. FUTURE WORK

The FL-COCOMO framework can be analyzed in terms of feasibility and acceptance in software industry. The framework can be extended to COCOMO II environment. Another extension can be use of Type-2 fuzzy to handle the uncertainty more precisely.

## 8. REFERENCES

- [1] Satyananda Reddy, KVSVN Raju, “ An Improved Fuzzy Approach for COCOMO’s Effort Estimation using Gaussian Membership Function”, Vol.(4), No. (5), July 2009.
- [2] Boehm, B.W., Software Engineering Economics, Prentice-Hall, Englewood Cliffs, NJ (1981).
- [3] Boehm B.W., Royce, W.W., Le COCOMO Ada, Genielogiciel & Systemes experts, 1989.
- [4] Boehm, B, W., et al, “Cost models foe future software life cycle processes: COCOMO2.0”, Annals of Software Engineering on Software process and Product Measurement, Amsterdam, 1995.
- [5] Walston, C.E., Felix, A.P, “A method of programming

- measurement and estimation”, IBM Systems Journal, Vol. 16, No. 1, 1997.
- [6] Putnam, L.H., “A general empirical solution to the macro software sizing and estimation problem”, IEEE Transaction on Software Engineering, No. 4, July 1978.
- [7] Jones, C., Programming Productivity, McGraw-Hill, New York, 1986.
- [8] Alberta’s., Gaffney, J.E., “Software function, Source lines of code, and development effort prediction: A Software science validation”, IEEE transactions on Software Engineering, Vol.SE-9, No. 6, Nov, 1983, pp 639-647.
- [9] Matson, J.E., Barrett, B.E., Mellichamp, J.M., “Software development cost estimation using function points”, IEEE Transactions on Software Engineering, Vol. 20, No.4, Apr., 1994, pp.275-287.
- [10] Leonard J.Jowers A, James J. Buckley B, Kevin D. Reilly A, “Estimation of f-COCOMO model parameters using optimization techniques”, <http://sunset.usc.edu/events/2006/CIIForum/pages/presentations/2006SEWorldJowers-Buckley-Reilly-c.pdf>.
- [11] J. Ryder, “Fuzzy modeling of software effort prediction”, Proceedings of IEEE Information Technology Conference, Syracuse, NY, 1998.
- [12] Z. Fei, and X. Liu, “f-COCOMO: fuzzy constructive cost model in software engineering”, Proceedings of the IEEE International Conference on Fuzzy Systems, IEEE Press, New York, 1992 pp. 331–337.
- [13] K. Srinivasan, and D. Fisher, “Machine learning approaches to estimating software development effort”, IEEE Transactions on Software Engineering, 21(2) 1995.
- [14] B. Boehm, C. Abts, and S. Chulani, “Software development cost estimation approaches—a survey”, Technical Reports, USC-CSE-2000505, University of Southern California Center for Software Engineering, 2000.
- [15] A. C. Hodgkinson, and P. W. Garratt, “A neurofuzzy cost estimator”, in: Proceedings of the Third International Conference on Software Engineering and Applications—SAE 1999, pp. 401–406.
- [16] C. Schofield, “Non-algorithmic effort estimation techniques”, Technical Reports, Department of Computing, Bournemouth University, England, TR98-01, March 1998.
- [17] C. Kirsopp, and M. J. Shepperd, “Making inferences with small numbers of training sets”, Sixth International Conference on Empirical Assessment & Evaluation in Software Engineering, Keele University, Staffordshire, UK, 2002.
- [18] A. Idri, and A. Abran, “COCOMO cost model using fuzzy logic”, Seventh International Conference on Fuzzy Theory and Technology, Atlantic City, NJ, 2000.
- [19] P. Musilek, W. Pedrycz, G. Succi, and M. Reformat, “Software cost estimation with fuzzy models”, Applied Computing Review, 8(2) 2000 24–29.
- [20] G. D. Boetticher, “An assessment of metric contribution in the construction of a neural network-based effort estimator”, Proceedings of Second International Workshop on Soft Computing Applied to Software Engineering, 2001.
- [21] C. J. Burgess, and M. Lefley, “Can genetic programming improve software effort estimation? A comparative evaluation”, Information and Software Technology, 43 2001, pp. 863–873.
- [22] M. Shepperd, and G. Kadoda, “Comparing software prediction techniques using simulation”, IEEE Transactions on Software Engineering, 27(11) 2001, pp. 1014–1022.
- [23] B. K. Clark, “The Effects of Software Process Maturity on Software Development Effort”, PhD Dissertation, Faculty of Graduate School, University of Southern California, 1997.
- [24] Harsh Kumar Verma, Vishal Sharma, “Handling Imprecision in Inputs using Fuzzy Logic to Predict Effort in Software Development”, IEEE 2nd International Advance Computing Conference, 2010, pp. 436-442.
- [25] Ch. Satyananda Reddy, KVSVN Raju, “Improving the accuracy of effort estimation through Fuzzy set combination of size and cost drivers”, WSEAS TRANSACTIONS on COMPUTERS, Vol(8), Issue (6), pp. 926-936, June 2009.