Volume 4, Issue 4, April 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Fuzzy Model for Function Point Analysis for Software Effort Estimation

Jaideep Kumar
Student, M.Tech, Department of Computer Science,
Subharti Institute of Engineering and Technology, Subhartipuram, Meerut, India

***Abstract-** As we all know that fuzzy logic has many successful applications both in industry and in academic research. It is noted that traditional estimation approaches can face serious difficulties when used on software engineering data, which is usually limited, incomplete and imprecisely collected. To resolve these problems a composite technique for building software models based on a mix of data, expert judgment and fuzzy logic is proposed. It is believed that the results reported here will stimulate and revitalize the study of the applicability of fuzzy logic to real- world software engineering problems. Here we will discuss techniques of estimation of various software attributes and then some new models/formulae are proposed to gain a better estimation of software attributes using fuzzy logic. It shows that fuzzy logic can be applied to estimate almost every software attribute, more accurately than non-fuzzy approaches. The selection of membership function, fuzziness in raw data and Defuzzification method used also affects the quality of estimation and precision of estimates.*

***Keywords:** Loc, Cocomo, Fpa, Ilf,*

I. INTRODUCTION

One of the most important problems faced by software developers and users is the prediction of the size of a programming system and its development effort. As an alternative to size, one might deal with a measure of the function that the software is to perform. Effort estimation process involves a series of systematic steps that provide estimate with acceptable risk. Software cost estimation starts at the proposal state and continues throughout the lifetime of a project. Out of the three principal components of cost i.e., hardware cost, training cost and effort cost, the effort cost is dominant.

It is observed that there is a need for effort prediction framework that can handle imprecision and uncertainty surrounding the prediction process and deal with the nature of inputs. In this paper, the focus is on effort estimation based on lines of code and function points. Two new models for effort estimation based on LOC and FP are proposed. For our study, triangular fuzzy numbers are used to represent size in terms of LOC and linguistic variables related to data and transactional functions. The developed models are tested on available experimental datasets.

FUZZY LOGIC

In 1948, Alan Turing wrote a paper, which marked the beginning of a new era, the era of the intelligent machine. To allow computers to mimic the way humans think, the theories of fuzzy sets and fuzzy logic was created. Classical logic deals with crisp knowledge where statements can only be either true or false, while fuzzy logic deals with vaguely formulated or uncertain knowledge.

FUZZY NUMBER

A fuzzy number is a quantity whose value is imprecise, rather than exact as in the case of ordinary single valued numbers. A fuzzy number is represented by a membership function, whose domain is a fuzzy set. The membership function associates a real number [0, 1] with each point in the fuzzy set, called degree of uncertainty or grade of membership. The membership µA (x) of an element x of a classical set A, as subset of the universe X, is defined by:

$$\mu_A(x) = \begin{cases} 1 \ iff \ \ x \in A \\ 0 \ iff \ \ x \notin A \end{cases} \quad \ldots(1)$$

In many respects, fuzzy numbers depict the physical world more realistically than single valued numbers. Suppose we are driving along a highway where the speed limit is 80km/hr, we try to hold the speed at exactly 80km/hr, but our car lacks cruise control, so the speed varies from moment to moment. If we note the instantaneous speed over a period of several minutes and then plot the result in rectangular coordinates, the resultant curve may look like one of the curves shown below; however, there is no restriction on the shape of the curve. The curves in Figure (1.1) show a triangular fuzzy number, a trapezoidal fuzzy number, and bell shaped fuzzy number.

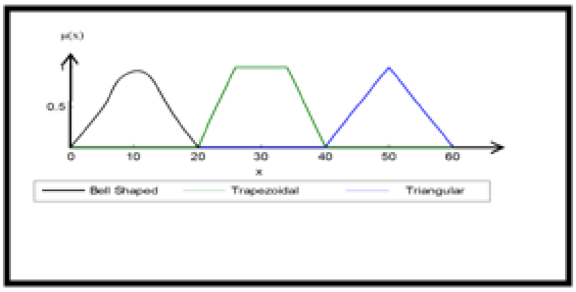

Figure 1.1: Membership Function

FUZZY LOGIC PROCESSES

The fuzzy logic process involves mainly following three sub-processes:

- Fuzzification
- Application of fuzzy logic
- Defuzzification

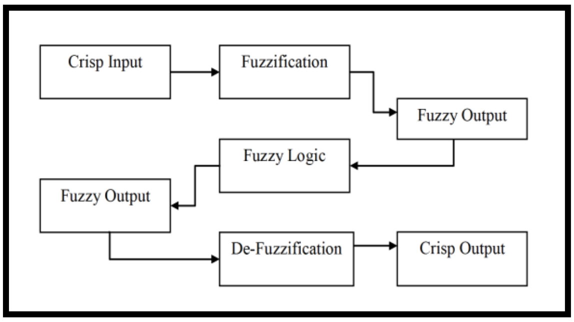

Figure 1.2: Illustrates the process of getting crisp out using fuzzy logic process.

Fuzzification: Fuzzification is a process whereby crisp values are converted to fuzzy values, represented by membership function. First step is to select appropriate linguistic/fuzzy system variables followed by defining fuzzy sets to represent concepts for each fuzzy variable. Assignment of fuzzy values may be done either by intuition or by some algorithm. Intuition involves contextual and semantic knowledge about an issue.

Application of fuzzy logic: Identification of rules to relate inputs to outputs, selecting techniques for correlation of inputs to outputs and composition of rules.

Defuzzification: Defuzzification is the process of converting fuzzy values to crisp values. Several techniques have been proposed for Defuzzification. The simplest method is the Maximum method in which a fuzzy set with maximum membership function is selected.

The rest of the paper is organized in five sections. Section II covers the basics of effort estimation techniques. Proposed models are presented in Section III. Section IV deals with experimental study and validation of results. Section V concludes the paper and discusses the experimental results.

II. BACKGROUND

The size of software will be the main factor that influences effort required in software development process. The size of the software can be measured with different methods; function points or counting source line of codes. These two sizing estimation techniques are discussed in this section.

A. Lines of Code (LOC)

A line of code, or LOC, appears to be the earliest measure used for Software Size. Early lines of code estimates are often based on expert opinion or analogy although empirical estimation methods also exist. In the year 1981, Bailey–Baisli [1] presented a model-generation process, which permits the development of a resource estimation model for any particular organization.

W. Pedrycz et al. [2] found that the concept of information granularity and fuzzy sets, in particular, plays an important role in making software cost estimation models more user-friendly. Ali Idri et al. [3] proposed the use of fuzzy sets in the COCOMO' 81 model. Musilek, P. et al.

[4] proposed f-COCOMO model, using fuzzy sets. They claim that methodology of fuzzy sets giving rise to f - COCOMO is sufficiently general to apply to other models of software cost estimation such as function point method.

Alaa F. Sheta et. al. [5], [6], introduced KLOC based effort estimation models using Genetic Algorithms and soft computing techniques.

B. Function Point Analysis (FPA)

Functionality measures are an alternative class of size measure. Albrecht [7] has developed a methodology to estimate the amount of the "function" the software is to perform, in terms of the data it is to use (absorb) and to generate (produce). The "function" is quantified as "function points," essentially, a weighted sum of the numbers of "inputs," "outputs," master files," and "inquiries" provided to, or generated by, the software.

IFPUG [8] has developed some standard terms and definitions related to function points, but these terms and definitions need to be applied to a variety of different software environments. The overall objective is to determine adjusted function point count. Function point analysis (FPA) begins with the decomposition of a project or application into its data and transactional functions.

The data functions, (ILF and EIF) represent the functionality provided to the user by attending to their internal and external requirements in relation to the data, whereas the transactional functions, (EI, EO and EQ) describe the functionality provided to the user in relation to the processing of this data by an application. Each function is classified according to its relative functional complexity as low, average or high.

The data functions relative functional complexity is based on the number of data element types (DETs) and the number of record element types (RETs). The transactional functions are classified according to the number of file types referenced (FTRs) and the number of DETs [9], [8].

The function points (FP) are evaluated in the following manner:

$$\text{Function Points} = \sum_{i=1}^{5} \sum_{j=1}^{3} \left(F_{ij} * Z_{ij}\right) * \mathrm{VA_dF} \quad \ldots 2.1$$

Where, Z_{ij} denotes count for component i at level j (low, average or high) F_{ij} is corresponding Function Points, VA_dF denotes Value Adjustment Factor it is derived from the sum of the degree of influence (DI) of the 14 general system characteristics (GSCc). The DI of each one of these characteristics ranges from 0 to 5.

Function points can be used to estimate effort required in software development. A numbers of studies have attempted to relate LOC and FP metrics [10]. The average number of source code statements per FP has been derived from historical data for numerous programming languages. Jones [11] gave programming language levels and Average numbers of source code statements per function point. The following abbreviations are used in the text follows:

ILF	Internal Logical Files
DET	Data Element Types
EIF	External Interface Files
RET	Record Element Types
EI	External Inputs
FTR	File Type Referenced
EO	External Outputs
EQ	External Query

III. PROPOSED APPROACH

Software project sizing is done in the project life cycle when the least is known and project uncertainty is greatest. Most of the traditional techniques such as function points, regression models, COCOMO, do not provide for uncertainty in their inputs, processing or outputs. New paradigms as Fuzzy Logic may offer an alternative for this challenge. Many of the problems of the existing effort estimation models can be solved by incorporating Fuzzy Logic. Even though some fuzzy logic based models exist in literature, yet there is a scope for more accurate estimation models using different approach for selection of linguistic variables and methods for Defuzzification.

This section introduces two new models for effort estimation based on fuzzy function points and fuzzy KLOC, which demonstrate the use of fuzzy logic as a means of capturing and reasoning with uncertainty in software effort estimation models.

A. LOC Based Model

A model based on some theoretical aspects related to linear-model structure development process is proposed. Adding the effect of Methodology (ME) will improve the model prediction quality. It was also found that adding a bias term similar to the classes of regression models helps to stabilize the model and reduce the effect of noise in measurements.

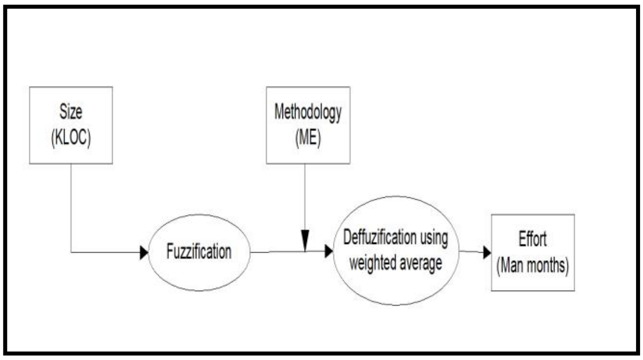

Figure 3.1: Proposed Framework of Fuzzy Effort Estimation Model (based on KLOC)

$$\text{Effort in ManMonths(MM)} = a(\text{KLOC})^b + c(ME) + d \qquad \text{....(3.1)}$$

Where, a, b, c and d are model parameters, the values of which are tuned, so that they are most suited to accurate estimation of software effort for project development.

Size of the project, especially in the beginning of the project, cannot be taken precisely. Size is taken as fuzzy number represented by TFN (α, m, β) and Defuzzification is carried out using weighted average method given in equation 3.3.

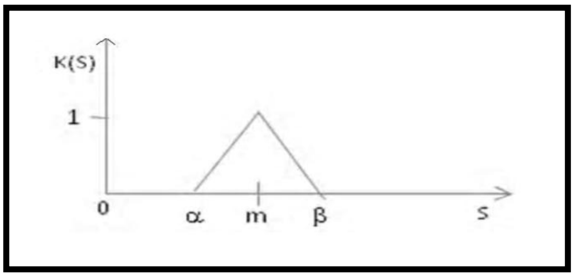

Figure 3.2: Representation of Fuzzy Number TFN (α, m, β)

$$\text{Effort} = \frac{w_1 E_a + w_2 E_m + w_3 E_\beta}{6} + e(ME) + d \qquad \text{....(3.2)}$$

Where, m is size estimate in KLOC, w1, w2 ,w3 are weights for optimistic (E), most likely (Em) and pessimistic estimates (E) respectively. We take, w1=1, w2=4, w3=5 i.e., we give highest credence to “most likely” estimate Em.

B. Fuzzy Functional Point Analysis (FFPA) Model

In this section, we use fuzzy logic technique to estimate size of software in terms of fuzzy functional points and hence the effort estimation in man-days. Triangular fuzzy numbers are used to represent the linguistic terms in Function Point Analysis (FPA). The use of linguistic terms allows the substitution of the classic intervals by fuzzy sets, creating an extension of the original model. The model simulates the way human being interpret linguistic terms; and the transition from one linguistic term to another adjacent one becomes gradual, unlike the original abrupt way. FFPA consists of the following stages:

- Modification of Complexity Matrices
- Fuzzification of Linguistic Variables
- Defuzzification

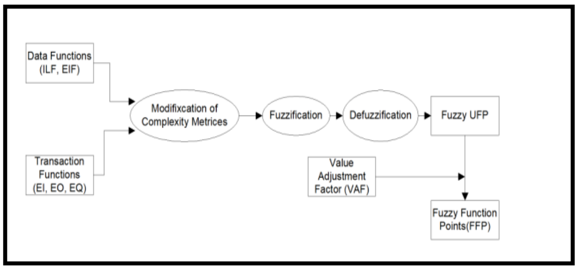

Figure 3.3: Proposed Framework of Fuzzy Function Points Estimation

Modification of Complexity Matrices

We take each linguistic variable as a triangular fuzzy numbers, TFN (α, m, β) where α <m< β. The membership function (x) for which is defined as:

$$\mu(x) = \begin{cases} 0, x \le \alpha \\ \dfrac{x-\alpha}{m-\alpha}, \alpha \le x \le m \\ \dfrac{\beta - x}{\beta - m}, m \le x \le \beta \\ 0, x \ge \beta \end{cases} \quad \text{....(3.3)}$$

We create a new linguistic variable, TFN (α, m, k), high or very high, where k is a positive integer. In case low and average are given, we create high variable. In case low, average and high are given, we create very high variable. The creation of the new linguistic variable helps to deal better with larger systems.

We can estimate the function points for the new variable, by extrapolation using Newton's interpolation formula. The extended function point counting weights are given in Table 3.1.

Table 3.1: Extended Function Point Counting Weights

Level	Function Point				
	ILF	EIF	EI	EO	EQ
Low	7	5	3	4	3
Avg	10	7	4	54	
High	15	10	6	7	6
Very High	22	14	9	10	9

IV. EXPERIMENTAL STUDY AND DATA ANALYSIS

A. Model Based on LOC

For this study I have taken data of randomly selected ten projects of NASA/SEL [1], [5].

The value of parameters a, b, c and d for proposed model, was obtained by tuning, for maximizing Variance Accounted for (VAF) and minimal Mean Absolute Relative Error (MARE). The estimated values of parameters are as follows:

a=3.3602; b= 0.8116; c= - 0 .4524 d=17 .8025

Effort in person-months is calculated using proposed fuzzy logic model and compared with available models. Validation results for proposed model are shown in Table 4.1.

Performance of proposed model is evaluated using VAF, MARE and Pred (25), as described in Table 4.1.

Table 4.1 Validation results of Proposed Model

Model	VAF	MARE	Pred. (25)
Bailey-Basility Model	93.15	14.74	90
Alaa, GA Model	98.93	44.75	60
Alaa,Fuzzy Model	98.87	10.64	90
ProposedFuzzy Model	99.04	10.63	90

B. Model Based on Function Points

Dataset for experimental study is taken from [12]. We choose an OLTP (On Line Transaction Processing) type of applications within a large ERP system running by a steel company implemented with COBOL on a database system. Three applications are studied viz. Project A, Project B and Project C.

To check the accuracy of proposed model in effort estimation, IFPUG function points need to be converted to man-days. Again, we use System B and System C as training data to find the average man-days per Function Point.

Table 4.2: Function Points (FP) and Actual Effort using Conventional Method

Project	FP	Actual Effort (Man-days)	Man-days/FP
B	400	126	0.315
C	319	95	0.298

The average man-days for a function point are therefore taken to be the average of the data from two projects B and C i.e. 0.307. The average man-days for a function point are used to estimate the man-days for project A. The function points by IFPUG method for project A are 111 and the estimated man-days are 34 (111*0.307), while the actual man-days for the project A are 32.

The result is compared that of the Bing's regression model [13] and proposed fuzzy model. The accuracies of proposed fuzzy logic based model is 99%, whereas the accuracy of effort estimation using conventional IFPUG method and Bing ' s Regression model are 94% and 95% respectively. Validation results for effort estimation models based on function points are depicted in Figure 4.1

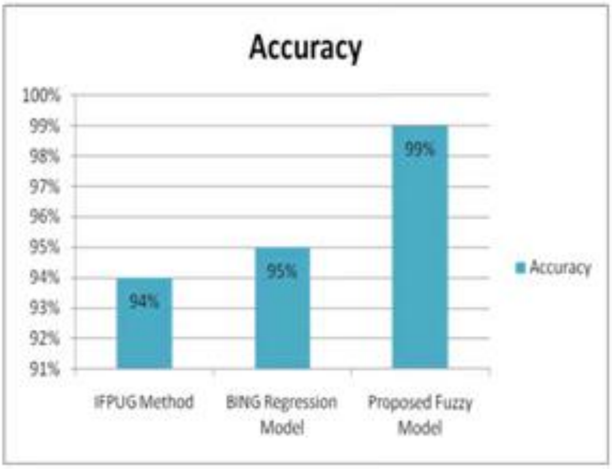

Figure 4.1: Accuracy of Effort Estimation Models (Based on FP)

V. CONCLUSION

In this paper, two new models for effort estimation based on LOC and functional point analyses (FPA) using fuzzy logic techniques are proposed. Triangular fuzzy numbers are used for linguistic terms. The results show remarkable improvement in the existing models and are sure to support in better software project management. The LOC based model considers size as a fuzzy number. Experimental study is conducted on NASA software projects dataset. Comparison of proposed model is carried out with the existing leading models. MARE for effort estimation by proposed model is about 10.63%, which is lower than that of existing models. In FPA, linguistic terms are used for some ranges of DET for which function points are considered same. Complexity matrices for various data and transaction functions are modified by using extension principle. Fuzzy numbers are used to represent DET ' s, with the help of which we get variation of function points throughout the range represented by a linguistic term to get estimation that is more accurate. Fuzzy function points evaluated are used for effort estimation in man- days. The proposed model is tested on an OLTP (On Line Transaction Processing) type of applications within a large ERP system running by a steel company. The experimental result shows that the accuracy of effort estimation in man-days using proposed model is 99% which is sufficiently higher than the estimate from available models. It is observed that the proposed model gives better results as compared to some earlier models. The approach used in this work represents a substantial departure from the conventional fuzzy logic techniques. The methodology of fuzzy sets used, is sufficiently general and can be applied to other areas of quantitative software engineering.

REFERENCES

[1] Bailey, J.W. and Basili,V.R., A Meta model for software development resource expenditure, Proc. Intl. Conf. Software Engineering, pp: 107-115, 1981.

[2] Pedrycz W., Peters J.F., Ramanna S., A Fuzzy Set Approach to Cost Estimation of Software Projects, Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering Shaw Conference Center, Edmonton Alberta, Canada. 1999.

[3] Ali Idri , Alain Abran and Laila Kjiri, COCOMO cost model using Fuzzy Logic, 7th International Conference on Fuzzy Theory & Technology Atlantic, New Jersy, March-2000.

[4] Musílek, P., Pedrycz, W., Succi, G., & Reformat, M., Software Cost Estimation with Fuzzy Models. ACM SIGAPP Applied Computing Review, 8(2), 24-29, 2000.

[5] Sheta, A. F., Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects, Journal of Computer Science 2(2):118-123, 2006.

[6] Sheta, A., Rine, D., Ayesh, A., Development of Software Effort and Schedule Estimation Models Using Soft Computing Techniques, IEEE World Congress on Computational Intelligence, 2008, Hong Kong, pp. 1283-1289,2008. [ALBR79] Albrecht A.J., Measuring application development productivity, SHARE/GUIDE IBM Application development Symposium, Monterey, California, pp 83-92, 1979.

[7] Albrecht A.J., AD/M Productivity Measurement and Estimate Validation-Draft, IBM Corp. Information Systems and Administration, AD/M Improvement Program, Purchase, NY, May 1, 1984

[8] Function point counting Practices: Manual Release 4.0, International Function Point User Group (IFPUG), Westerville, Ohio, 1994.

[9] Sprouls, J., IFPUG Function Point Counting Practices Manual, Release 3.0, Westerville, Ohio: International Function Point Users Group, 1990.

[10] Lima, O.S.J., Farias P.P.M. and Belchio A.D., " A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessments " , CLEI Electronic Journal, Volume 5: Number 2, January 2002.

[11] Jones, C., Programming Languages Table, Release 8.2, March 1996, available at http://www.spr.com/library/0langtbl.htm.

[12] Basili V. R., Freburger K., Programming Measurement and Estimation in the Software Engineering Laboratory, Journal of System and Software, 2, 2, 47-57, 1981.

[13] Bingchiang Jeng, Dowming Yeh, Deron Wang, Shu-Lan Chu, An Empirical Study of Effort Estimation Using Simplified Functional Point Analysis, International Computer Symposium, Dec 4-6, Taipei, Taiwan, 2006.

