

Fuzzy based Effort Estimation Approach

Vishal Chandra
AI, SGVU
Jaipur, Rajasthan, India

Savita Shiwani
IT, SGVU
Jaipur, Rajasthan, India

ABSTRACT

Besides of these there are many equation based effort estimation techniques like *Halstead Model*, *Bailey-Basil Model*, and *Walston-Felix Model*.

Cost and effort estimation are the major concern of any software industry. They are identified with the help of Kilo Line Code (KLOC) which denotes number of, line of code in software. For example if a software contain 2000 lines then it has 2 kilo line of code. There are several ways to estimate it with various pros and cons. We can categorize them in two parts, one is model based and another is equation based estimation technique.

In this paper a KLOC formula is proposed that is based on effort equation which is integrated with fuzzy logic to estimate effort. These components are cumulative, vague and fuzzy easily handles them as well. Various fuzzy membership functions are used in this paper.

One of the most popular model, based upon estimation technique is COCOMO and its variants, which is highly used in the industry, along with the other variants such as neuro fuzzy approach, fuzzy approach, and cost driver based estimations.

There are huge differences between Model Based and Equation Based effort estimation techniques, model based built on specific model, like architecture & available resources whereas equation based techniques follows some back- ground equations.

Keywords

Equation based, fuzzy logic, cumulative experience, effort estimation, membership functions, KLOC.

1. INTRODUCTION

Software engineers always try to estimate cost & effort with an accuracy in the software industry. One of the major target of the software engineers' community is, to develop useful models that can explain the SDLC and precisely predict the effort. There are many parametric software estimation models had been developed over the last three- four decades , that were based on efforts estimation like Jones and Software Productivity Research's [2], Checkpoint model, Putnam and Quantitative Software Measurement's [8], SLIM model, Park and PRICE Systems' PRICE-S model, Jensen and the SEER SEM model, Rubin and the Estimacs model and Boehm and the COCOMO model [Putnam, 1992, Jones, 1997, Park, 1988, Jensen, 1983, Rubin, 1983, Boehm, 1981, Boehm, 1995, Walkerden, 1997, Conte, 1986, Fenton, 1991, Masters, 1985, Mohanty, 1981].. These approaches impose a few restrictions, often violated by software engineering data and resulted in the development of inaccurate empirical models that do not perform very well when used for prediction of effort. This paper focuses on approximate effort estimation with the help of equation which is related to kilo line of codes (KLOC) and fuzzy multiplier.

This paper uses NASA project data set to validate the proposed model based on Cumulative experience containing following properties:-

- Programmer Qualifications
- Programmer Experience with Machine
- Programmer Experience with Language
- Programmer Experience with Application
- Working experience of programmer

These are vague things or properties which cannot be measured in terms of traditional mathematical crisp theory. Fuzzy works between two rigid boundaries between 0 and 1 or between yes and no. There are various factors in which effort estimation depends, in this paper only few (five) common properties are taken. This model is very useful in estimating medium and large projects which have more KLOC.

2. LITRETURE SURVEY

In last few decades many effort estimation techniques are developed. Software estimation techniques are broadly categorized in six categories they are as follows

- Model based
- Expertise based
- Learning oriented
- Dynamic Based
- Regression Based
- Composite

1. **Halstead Model [3].** This model developed by Halstead between delivered lines of source code and formulates a relation

$$\text{EFFORT} = 0.7 \times (\text{KLOC})^{1.15} \quad (1)$$

2. **Bailey-Basil Model [2].** This model developed by Bailey-Basil between delivered lines of source code and formulates a relation

$$\text{EFFORT} = 5.5 \times (\text{KLOC})^{1.16} \quad (2)$$

3. **Walston-Felix Model:** - Walston and Felix (1977) developed their effort model from a various aspects of the software development environment such as user database of sixty projects collected in IBM's Federal Systems division. It provides a relationship between delivered lines of source code. This model constitutes participation, customer-oriented changes, memory constraints etc. [6].

According to Walston and Felix model, effort is computed by

$$\text{EFFORT} = 5.2 \times (\text{KLOC})^{0.91} \quad (3)$$

$$\text{Duration } D = 4.1 \times (\text{KLOC})^{0.36} \quad (4)$$

4. **SEL – Model:** - The Software Engineering Laboratory (SEL) of the University of Maryland has established a model i.e. SEL Model for estimation. Estimation of effort according to SEL model is defined as follows:-

$$\text{EFFORT} = 1.4 \times (\text{KLOC})^{0.26} \quad (5)$$

$$\text{Duration} = 4.6 \times (\text{KLOC})^{0.26} \quad (6)$$

Effort (Person-Months) and lines of code (size in thousands of lines of code i.e. KLOC) are used as predictors.

5. **Doty (for KLOC > 9):-** This model developed by Doty between delivered lines of source code and formulates a relation.

$$\text{EFFORT} = 5.288 \times (\text{KLOC})^{1.047} \quad (7)$$

6. **COCOMO II model** it is a collection of three variants, Application composition model, early design model, and Post architecture model. This is an extension of intermediate COCOMO model and defined as: -

$$\text{EFFORT} = 2.9 \times (\text{KLOC})^{1.10} \quad (8)$$

3. PROPOSED MODEL

This paper is highly motivated by Bailey-Basil Model, which contain KLOC and multiplying factor. This paper proposed an empirical equation for estimating effort with the help of KLOC and fuzzy multipliers. Fuzzy multipliers are based on cumulative experiences such as

- Programmer Qualifications
- Programmer Experience with Machine
- Programmer Experience with Language
- Programmer Experience with Application
- Working experience of programmer

The proposed equation is

$$\text{EFFORT} = 0.73 \times (\text{KLOC})^{1.13} \times (1.01)^{F\text{-factor}} \quad (9)$$

Where KLOC is kilo line of codes and F-factor is fuzzy factor which is output of fuzzy expert system depending upon various above mentioned cumulative experiences. These cumulative experiences are scaled from 0 to 5.

Advantages

- Fit for any type of project.
- Based on history
- Repeatable
- Unique adjustment factors
- Has different modes
- Works well on similar projects
- Highly calibrated
- Well-documented
- Easy to use it.

Disadvantages

- There are huge amount of factor (vague) affecting effort.

- Difficult to analyze team.
- It based on intuitive judgment, it may be biased
- Ignores requirements volatility

1. **Programmer Qualification-** In the effort estimation programmers' qualification is important.

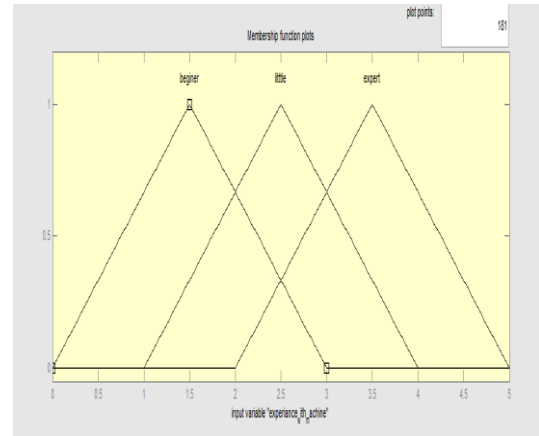


Figure 1. Programmer Qualification

Qualification reflects the skill of the employees Usually Higher qualified means more knowledge. It is assumed bachelor degree is the minimum degree for any job in software industry. 0 to 3 points are assigned for bachelor degree, 1 to 4 for post graduate degree and 2.5 to 5 for PhD and further qualification. It is assumed that more knowledge will decrease the effort applied.

2. **Programmer Experience with Machine:** - For the estimation of effort, programmers' experience with the machine is important to develop software project. If someone has expertise on a particular machine then effort will be less than beginner who have little knowledge of working machine. Proposed model assumed that beginner has every small knowledge of machine, this paper used the minimum in the 0 to 5 scale. In this paper it is assumed that Beginner starts with 0 and ends at 3. Employee who has little experience of machine related to his or her work lies between 1 and 4.

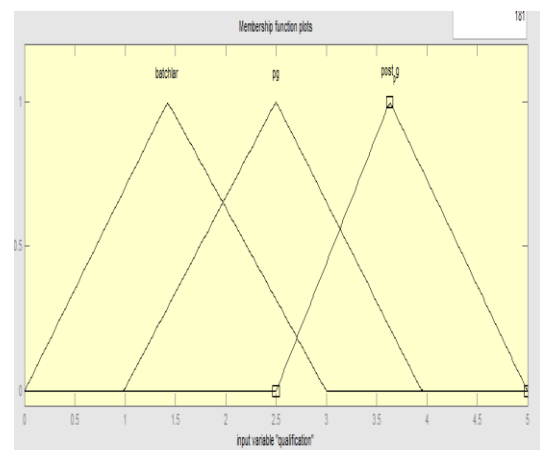


Figure 2. Programmers' experience with machine

3. **Programmer Experience with Language:** In the estimation of effort required for developing project, programmer experience with the language is important. Programmer's experience with language means

programming language by which he or she develop any application whether it is web application or standalone application in this paper it is assumed that novice which has very little knowledge assigned as 0 to 4, good between 1.5 to 3.5 and expert as 3 to 5.

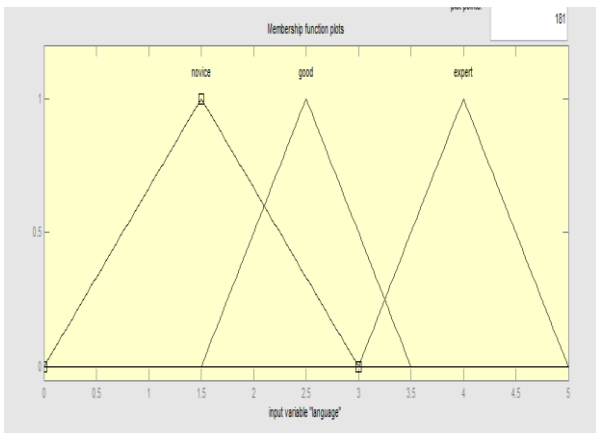


Figure 3. programmers' experience with language

- Programmer Experience with Application:** - significance of programmer's experience with application, through which software is going to developed is important during software development. Such as compilers, IDEs. In this paper It is assumed 0 to 2 as little experience with application, 1 to 3 as good and 2.5 to 5 as expert with the application.

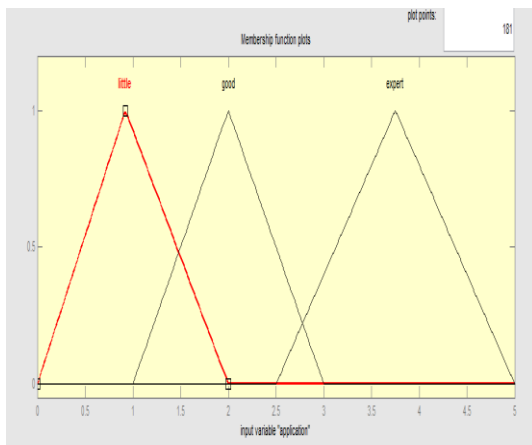


Figure 4. Programmers' Experience with Application

- Working experience of programmer:** - one of the most important factor for effort estimation is considering working experience of programmer. More experience means less effort and less experience more effort. In this paper it is assumed that 0 to 3 assigned as few experience, 5 to 3 as more experience, and 2 to 5 as many year experience.

Table 1. Software Projects Data			
Project	KLOOOC	Measured effort	calculated
1	90.2000	115.8000	120.29
2	46.2000	96.0000	56.34
3	46.5000	79.0000	58.0
4	54.5000	90.8000	69.09
5	31.1000	39.6000	37.10
6	67.5000	98.4000	88.00
7	12.8000	18.9000	12.8
8	10.5000	10.3000	10.6
9	21.5000	28.5000	23.9
10	3.1000	7.0000	2.6
11	4.2000	9.0000	3.6
12	7.8000	7.3000	6.97
13	2.1000	5.0000	1.6
14	5.0000	8.4000	4.7
15	78.6000	98.7000	103.3
16	9.7000	15.6000	9.1
17	12.5000	23.9000	12.5
18	100.8000	138.3000	139.6

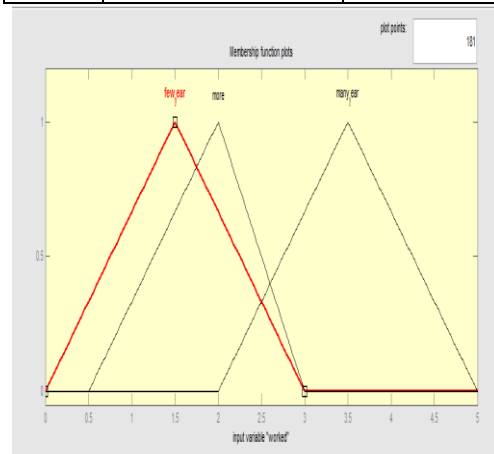


Figure 5 Working experience of programmer.

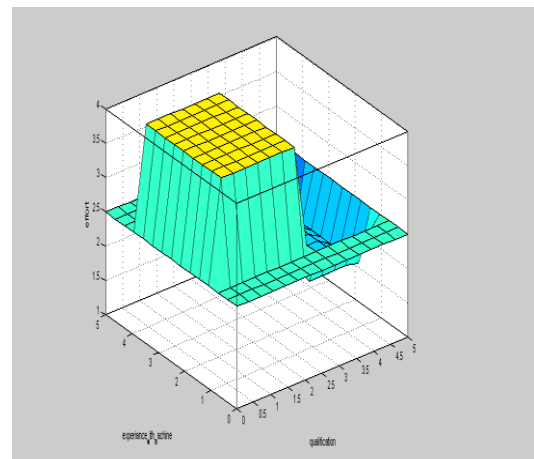


Figure 6 Surface Curve

4. CONCLUSION AND FUTURE SCOPE

This paper tried to solve effort estimation problem in software industry using fuzzy and cumulative approach. To validate the proposed model, NASA data is used. Equation gives result nearly within measured effort in NASA dataset used KLOC and cumulative experience as input and equation gives output as effort. This paper used few number of cumulative experiences, there may be possible many other cumulative experiences, also large dataset may be used to validate the equation. This model is well suited for medium and large amount of KLOC or large or medium software projects. There may be possible to enhance and apply some constraint to membership function.

5. REFERENCES

- [1] Baik, J. (2000) 'The effect of case tools on software development effort', PhD thesis, University of Southern California, CA, USA.
- [2] Bailey, J.W. and Basili, V.R. (1981) 'A meta model for software Development resource expenditure', in Proceedings of The International Conference on Software Engineering Pp.107–115.
- [3] Benediktsson, O., Dalcher, D., Reed, K. and Woodman, M. (2003) 'COCOMO based effort estimation for iterative and Incremental software development', *Software Quality Journal*, Vol. 11, pp.265–281. Birge, B. (2005) PSOT, Particle Swarm Optimization Toolbox for Matlab, web site www.matworks.com.
- [4] Boehm, B. (1981) *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ. Boehm, B. (1995) 'Cost models for future software life cycle process: COCOMO2', *Annals of Software Engineering*.
- [5] Boehm, B., et al. (2000) *Software Cost Estimation with COCOMOII*, Prentice Hall PTR. Boraso, M., Montangero, C. and Sedehi, H. (1996) *Software Cost estimation*.
- [6] *Estimation: An Experimental Study of Model Performances*, Technical report.
- [7] Briand, L.C., Emam, K.E. and Wiczorek, I. (1999) 'Explaining The cost of European space and military projects', in ICSE 99: Proceedings of the 21st International Conference on Software Engineering, pp.303–312, IEEE Computer Society Press, Los Alamitos, CA, USA.
- [8] Chulani, S. and Boehm, B. (1999) *Modeling Software Defect*.
- [9] Introduction and Removal: Coqualmo (Constructive Quality Model), Technical report, Technical Report USC-CSE-99-b510, University of Southern California, Center for Software Engineering.
- [10] Chulani, S., Boehm, B. and Steece, B. (1999) 'Calibrating software cost models using Bayesian analysis', *IEEE Trans. Software*
- [11] Engr., July–August 1999, pp.573–583. Clark, B., Devnani-Chulani, S. and Boehm, B. (1998) 'Calibrating the COCOMO II post-architecture model', in Proceedings of the 20th International Conference on Software Engineering (ICSE '98), pp.477–480, IEEE Computer Society, Washington, DC, USA.
- [12] Dolado, C.J. and Lefley, M. (2001) 'Can genetic programming.
- [13] Improve software effort estimation? A comparative Evaluation', *Information and Software Technology*, Vol. 43, Pp.863–873.
- [14] Dubois, D. and Prade, H. (1992) 'Fuzzy sets in approximate Reasoning: part 1', *Fuzzy Sets and Systems*, Vol. 40, Pp.143–202. Group, T.S. (1995) 'CHAOS chronicles', PhD thesis, Standish Group Internet Report.
- [15] Gustafson, D.E. and Kessel, W.C. (1979) 'Fuzzy clustering with a fuzzy covariance matrix', in Proceedings of the IEEE CDC, San Diego, CA, USA, pp.761–766.
- [16] Hodgkinson, A.C. and Garratt, P.W. (1999) 'A neurofuzzy cost Estimator', in Proceedings of the Third Conference on Software Engineering and Applications, pp.401–406. Kemere, C.F. (1987) 'An empirical validation of software Cost estimation models', *Communication ACM*, Vol. 30, Pp.416–429.
- [17] Kosko, B. (1998) 'Fuzzy systems as universal approximators', in Proceedings of Int. Conf. Fuzzy Syst., Pp.1153.
- [18] Hodgkinson, A.C. and Garratt, P.W. (1999) 'A neurofuzzy costEstimator', in *Proceedings of the Third Conference on*
- [19] *Software Engineering and Applications*, pp.401–406. Kemere, C.F. (1987) 'An empirical validation of software Cost estimation models', *Communication ACM*, Vol. 30, Pp.416–429.
- [20] Kosko, B. (1998) 'Fuzzy systems as universal approximators', in *Proceedings of Int. Conf. Fuzzy Syst.*, Pp.1153.