

Software Effort Estimation: A Fuzzy Logic Approach

Vishal Chandra
AI, SGVU
Jaipur, Rajasthan, India

ABSTRACT

There are many equation based effort estimation models like Bailey-Basil Model, Halstead Model, and Walston-Felix Model. Effort and Cost estimation are the major concern of any sort of software industry. They are calculated with the help of Kilo Line Code (KLOC) which are the number of, line of code in software. For example if a software contain 1000 lines then it has 1 kilo line of codes. There are several ways to estimate it with various advantages and disadvantages. We can distinguish them in two parts, one is equation based and another is model based estimation techniques.

In this paper a KLOC formula is proposed that calculate effort which is integrated with fuzzy logic. These components are vague and fuzzy easily handles them as well. Various fuzzy triangular membership functions are used in this paper.

One of the most popular model is COCOMO [5] and its variants, which is mostly used in the industry, along with the other variants such as fuzzy approach, neuro fuzzy approach and cost driver based estimations.

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

Keywords

Empirical Equation, fuzzy logic, effort estimation, membership functions, KLOC.

1. INTRODUCTION

Software engineers try to estimate effort & cost with an accuracy in the software industry. The major target of the software engineering community is to develop useful models that can explain precisely predict the effort. There are many empirical equation based software estimation models had been developed over the last two- 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[1] 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 fuzzy parameters, there are many

fuzzy parameters but I have chosen six they are as follows[8].:-

- Qualification of Programmer
- Experience with machine
- Experience with Language
- Experience with Application
- Working experience of programmer
- Complexity
- Storage
- Required reuse
- Facilities
- Development schedule
- Product reliability and complexity
- Analytical capability

These are vague properties which cannot be measured in terms of traditional mathematical theory or crisp theory. Fuzzy works between two rigid boundaries which we say 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 small, medium and large projects which have any amount of KLOC.

2. LITRETURE SURVEY

In last few decades a large number of software development effort estimation techniques are developed. Software estimation techniques are broadly categorized in six categories they are as follows [10].

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

2.1 Halstead Model [3]. This model was developed by Halstead. Source code and formulates a relation [4]

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

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

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

2.3 Walston-Felix Model: - Walton 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

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

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

2.4 SEL – Model: - The Software Engineering Laboratory (SEL) of the University of 'Maryland' has established a model [8] i.e. SEL Model for estimation. Effort according to SEL model is defined as follows:-

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

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

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

2.5 Doty (for KLOC > 9):- This model developed by Doty source code and formulates a relation.

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

2.6 COCOMO mode [3] it is a collection of three variants, organic, inorganic and semidetached. Application composition model, early design model, and Post architecture model. This is an extension of intermediate COCOMO model and defined as: -

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

3. PROPOSED MODEL

This paper is highly motivated by COCOMO Model, which contain KLOC and Fuzzy factor (F-factor).

This paper proposed an empirical formula for estimating effort with the help of KLOC and fuzzy factor for different type of team. Fuzzy multipliers are based on vague parameter or effort drivers such as [2]

- Qualification of Programmer
- Experience with machine
- Experience with Language
- Experience with Application
- Working experience of programmer
- Complexity
- Storage
- Required reuse
- Facilities
- Development schedule
- Product reliability and complexity
- Analytical capability
- Training given

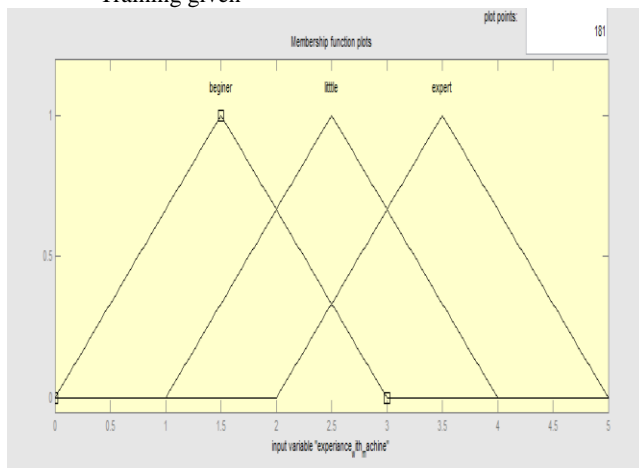


Figure 1. Qualification of Programmer

The proposed equation is

$$\text{EFFORT} = a(KLOC)^b \text{ PM} \quad (9)$$

Team	F-factor	a	b
Beginner	1-2.5	0.6	1.17
Experienced	2.5-4	0.6	1.16
Expert	4-5	0.6	1.14

A team in which members are low qualified, less experienced and many more properties have to do high effort whereas team in which members have high qualification, experience with language and machine required less effort to developed software.

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 parameters. These vague parameters are scaled from 0 to 5.

F-factor is the output of parametric fuzzy expert system.

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
- Ignores documentation
- Oversimplifies security
- Ignores software safety
- Ignores many hardware issues
- Personnel experience may be obsolete
- Must know the vague properties
- Must be able to predict project size

To validate proposed empirical equation this paper used some vague experiences, on which we can calculate and verify effort

3.1 Qualification of Programmer

For effort estimation, qualification of programmers or developers is important.

Qualification shows the mental skill of the employees Usually Higher qualification means more knowledge, more skill, better understanding of problem. In this paper it is assumed that bachelor degree is the minimum qualification for any job in software industry. 0 to 3 points are given 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 given to developing software.

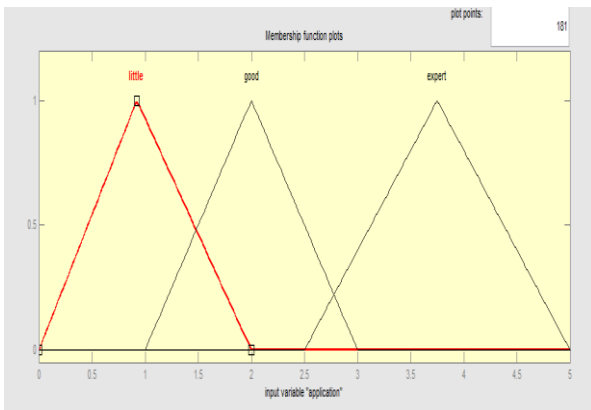


Figure 2. Programmers' experience with machine

3.2 Experiences with Machine

Estimation of effort, experience of programmer with the machine is important for develop software project. If someone has sound knowledge on a particular machine then he or she required less effort than beginner who have less knowledge of working with machine. In this paper it is assumed that beginner has every small knowledge of machine, proposed model 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.

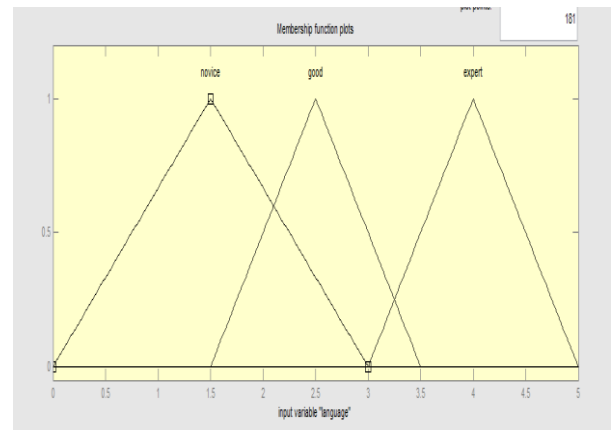


Figure 4. Programmers' Experience with Application

3.4 Experiences with Application

Programmer's or developers' experience with software or application which is used to develop software is important during software development. Such as compilers, IDEs. 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.

3.5 Working Experience

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.

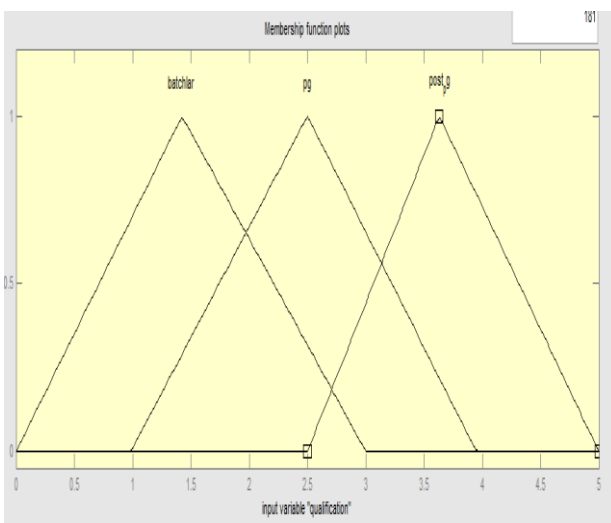


Figure 3. Programmers' experience with language

3.3 Language Experience

For the development of any software or program, programming language required. For the estimation of effort required for developing software project, programmer experience with the language is important. 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.

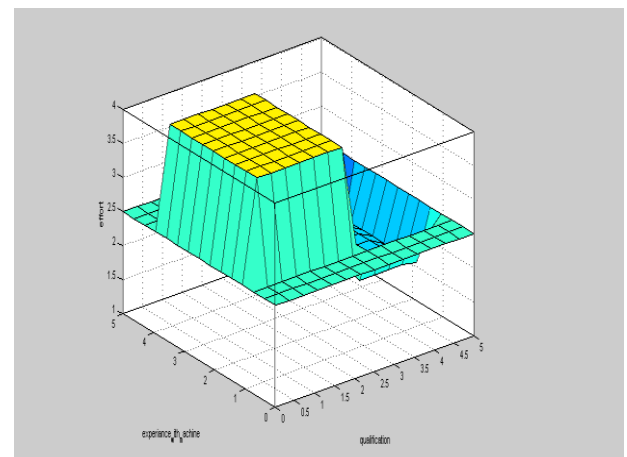


Figure 5 Working experience of programmer

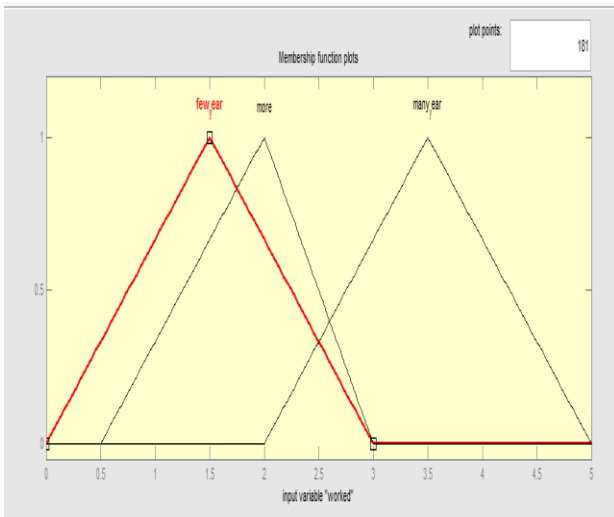


Figure 6. Surface Curve

4. CONCLUSION AND FUTURE SCOPE

This paper tried to solve effort estimation problem in software industry using fuzzy and vague property in which effort depends. For the validation of the proposed model, NASA data is used. Proposed Equation gives result nearly measured effort in NASA dataset used KLOC and cumulative experience as input and equation gives output as effort. This paper used few number of vague parameters, there may be possible many other properties, also large dataset may be used to validate the equation. This model is well suited for small, medium and large software projects. There may be possible to enhance and apply some constraint to membership function.

“The models are just there to help, not to make the management decisions for you.”[10]

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: An Experimental Study of Model Performances, Technical report.
- [6] 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.
- [7] Chulani, S. and Boehm, B. (1999) Modeling Software Defect Introduction and Removal: Coqualmo (Constructive QualitybModel), Technical report, Technical Report USC-CSE-99-b510, University of Southern California, Center for Software Engineering.
- [8] Chulani, S., Boehm, B. and Steece, B. (1999) ‘Calibrating software cost models using Bayesian analysis’, IEEE Trans. Software Engr., July–August 1999, pp.573–583.

Table 1. SOFTWARE PROJECTS		
DATA		
Project no.	KLOC	Measured effort calculated
1	90.1000	115.8000 121.29
2	47.2000	96.0000 56.34
3	46.00	79.0000 58.0
4	55.5000	90.8000 69.09
5	30.1000	39.6000 37.10
6	68.5000	98.4000 88.00
7	13.8000	18.9000 12.8
8	11.5000	10.3000 10.6
9	20.5000	28.5000 23.9
10	4.1000	7.0000 2.6
11	4.200	9.0000 3.6
12	8.8000	7.3000 6.97
13	6.16000	5.0000 1.6
14	7.0090	8.4000 4.7
15	78.6000	98.7000 103.3
16	8.7000	15.6000 9.1
17	12.5000	23.9000 12.5
18	100.8	138.3000 139.6

- [9] 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.
- [10] Dolado, C.J. and Lefley, M. (2001) 'Can genetic programming improve software effort estimation? A comparative Evaluation', Information and Software Technology, Vol. 43, Pp.863–873.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] Kosko, B. (1998) 'Fuzzy systems as universal approximators', in Proceedings of Int. Conf. Fuzzy Syst., Pp.1153.