International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 5, Issue 5
May 2018

Comparative Study of Software Effort Estimation Models

Nancy Sharma
Thapar Institute of Engineering and Technology, Patiala, Punjab (India)
Vineeta Bassi
Thapar Institute of Engineering and Technology, Patiala, Punjab (India)

ABSTRACT

Software is developed favorably only when software is estimated accurately. A number of software projects being developed since ages fail, just because of either overestimation or underestimation in terms of budget and time delivery. These two measures along with size are the critical measures for accurate effort estimation. An acute problem confronted by software project managers is to find accurate Software Effort Estimation (SEE) using efficient technique. Hence, it is very important to estimate effort and time in the beginning of the project development so that the developing team can attain some confidence to plan the project in advance and latter to deliver the quality product. In this paper, the main goal is to study the effort estimation using different approaches. Main focus of this study is on classical approach COCOMO, whose effort estimation is marked as a benchmark in the history of Software effort estimation and other approaches discussed in the paper are on the basis of KLOC formulae.

KEYWORDS: Software Effort Estimation (SEE), Constructive Cost Model (COCOMO), Kilo Lines of Code (KLOC), Person-months, Graphical User Interface (GUI), Effort metric.

1. INTRODUCTION

Software Engineering is a branch of computer science, which deals with the development of a software; considering former approaches, principles, procedures and experiences [1]. The end product is developed allowing for many aspects such as effort, cost, time, people and quality. In order to get a good end product, software project planning needs to be done, which includes main component of the planning i.e. Software Estimation. Software Estimation consists of following steps:

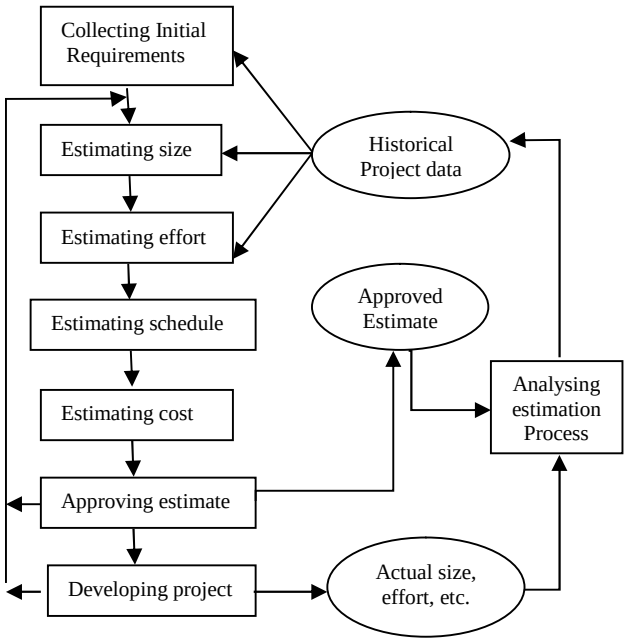

Fig. 1: Basic Project Estimation Process [2]

As shown in Fig. 1, basic project estimation process begins with collecting initial requirements in terms of resources like tools, techniques and historical data. The main procedure begins with estimating size considering customer's needs to be in the software product. Followed by estimating the effort in person months or so, which helps in calculating schedule and cost of the project. As the effort increases, the cost will also, increase [3]. Entire estimation is done using estimation tools. Estimations are approved in the later stage and the process to software development starts at this point. All the estimated data gets stored and is compared with the actual units of effort, cost and size etc.

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 5, Issue 5
May 2018

Among all the estimates, SEE plays a vital role in the development of software[4]. SEE is the procedure of auspicating the real effort; mostly calculated in person hours or person months, in order to develop the software. The models used to estimate effort attain the particular procedural criteria briefly described below:

1.1. SOFTWARE EFFORT ESTIMATION MODELS

Various SEE models came into existence when people started following proper project management process. Lately a survey on SEE explored that people often use KLOC, Story Points, Function Points and Use Case Points as a measure of size [5].

There are a number of techniques used to calculate effort namely, techniques for machine learning, regression, pricing to win, algorithmic models, Parkinson's law, estimation by analogy, top-down approach, bottom-up approach and expert judgment are generally accessed for software effort estimation.SEE is being calculated by using a number of models like [6]:

- COCOMO
- COCOMO II
- Halstead
- Bailey-Basili
- Software Engineering Laboratory (SEL)

1.1.1 COCOMO

Widely famous effort estimation approach, Constructive Cost Model also known as COCOMO, is considered as one of the best approaches since 1981 when Bary Boehm introduced it to the world. Effort and cost are computed using COCOMO model as a function of the number of Lines of Code (LOC) and KLOC (LOC and KLOC that are estimated and are coded for a project) [7].

$$E = a(KLOC)^b \quad (1)$$

Where,

E represents Effort in person months. a and b are the constants and their values are related to technical nature and environment of the development of system that depends on the class of the software project. COCOMO generally refers to a group of models. There are three modes into which a software project has been classified by Boehm, depending upon the complexity of the project. These modes are: Organic, Semi-Detached, Embedded [8].

Organic: These projects are small level projects, developed by small teams and at a non-rigid in-house like environment.

Semi-Detached: These projects are developed by the teams having medium level of experience with less than tough constraints. It is the combination of Organic and Embedded modes.

Embedded: These type of projects being developed have complex constraints and if changes occur in the projects in the later stage, then it becomes very costly to solve those issues.

Following table briefs the Effort Estimation formulae for basic COCOMO's models.

Table 1: Basic COCOMO Models [9]

Category	a	b	Effort(E)
Organic	2.4	1.05	$E=2.4(KLOC)^{1.05}$
Semi-Detached	3.0	1.12	$E=3.0(KLOC)^{1.12}$
Embedded	3.6	1.20	$E=3.6(KLOC)^{1.20}$

1.1.2. COCOMO II

COCOMO II is the updated version of COCOMO, which overcomes the weakness of COCOMO to estimate efforts of newer developed software. COCOMO II is good at allowing accuracy as 20% in cost and 70% in time [10].

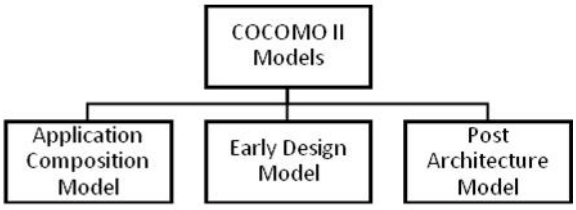

Fig. 2: COCOMO II models

Application Composition Model: This model is for those projects, which have latest GUI-builder tools.

Early Design Model: Before determining the actual architecture only, this model provides the facility in which one can estimate the cost and time duration of the project in rough manner.

Post Architecture Model: This model is applied when final plan of entire architecture of the project

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 5, Issue 5
May 2018

development is available. It takes the size of project in Kilo Source Lines of Code (KSLOC) [11].

Effort can be calculated using given formula:

$$E=2.9 * (KLOC)^{1.10} \quad (2)$$

1.1.3. Halstead's Model

M.H. Halstead contributed his work in software metrics in 1977 and gave a software science to estimate effort for the development of the project. He stated metrics, which are dependent on the unique number of operators and operand counts.Hypothetical form of Halstead's metric says that the effort (E) is directly proportional to the product of volume of the program (V) and level of difficulty of the program (D)[12].

Effort in Halstead model can be calculated using given formula:

$$E=V*D \quad (3)$$

In terms of KLOC,

$$E=5.2(KLOC)^{1.50} \quad (4)$$

1.1.4. Bailey-Basili Model

Bailey& Basili developed a model of effort estimationby hypothesizing the mode of reducing the standard error of estimation with the help of non linear least square regression [13]. The model considers the cost drivers such as experience, complexity and methodology to improve typical estimation.

Effort in Bailey-Basili modelcan be calculated using given formula:

$$E=5.5+0.73(KLOC)^{1.16} \quad (5)$$

1.1.5. Software Engineering Laboratory (SEL) Model

SEL model of estimation is developed by the University of Maryland [14].

Effort in SEL model can be calculated using given formula:

$$E=1.4 * (KLOC)^{0.93} \quad (6)$$

Effort is in Person Months and size is in thousand lines of code i.e. KLOC are used as units.

2. LITERATURE REVIEW

C. Thirumalai et.al (2017),have analyzed the effect of Halstead's product hypothesis on Python programming language using Halstead's metric. A Python program has been used to calculate effort using Halstead's metric and the output value varies from actual time[15]. Likewise, a Java program has been analyzed and compared using Actual effort, Halstead's metric and compared with COCOMO II.The results in the form of Strout's standard value, show that Halstead's metric predicts the actual time and effort precisely for Python Program as well as for Java program.

M. Madheswaran et.al (2014), have constructed a model to estimate effort and cost with the help of artificial neural networks, in which COCOMO model gets mapped to neural network and that can enhance the predication accuracy using neural network in COCOMO model for software project[16]. A multilayer feed forward neural network has been used which contains identifying the function at three units namely, input, hidden and output.For the network training purpose, back propagation algorithm is used along with COCOMO dataset. The prediction of Software development effort is based on actual effort, effort calculated using COCOMO model and neural networks model.

Phu Le et.al (2017), have compared four models of effort estimation namely, actual COCOMO, COCOMOcalibration, k-nearest neighbors and a combination of k-nearest neighbors and COCOMO calibration; and have proved that a combination of k-nearest neighbors and COCOMO calibration performs better among others, only when small set of nearest neighboring projects are taken[17].It has been concluded that worst or best performance of this model is based on the increase and decrease of datasets respectively.The k-nearest neighbors model is resulted to be in worst performance and provides more estimation errors, when there is the use of weighted average of nearest neighboring projects. The comparison is based on the values of Mean Absolute Residual (MAR) and Mean Magnitude of Relative Error (MMRE).

T. Menzies et. al (2016), have investigated the results in the form of calculated effort from newer as well as older SEE methods to find out that the way of collecting of data is more crucial over the impact of technique that has been applied on that data [18].In five steps, the comparison has been made. First of all, methods from past to modern era

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 5, Issue 5
May 2018

i.e. from COCOMO to analogical reasoning type of methods were collected. Secondly, record of issue leading to the need of new methods when there was COCOMO already been used for SEE. In third step, collation between SEE methods based on the classification of the issues collected has been done. Fourth step contains performing experiments on COCOMO's 1991,2000,2005 and 2010 datasets. In the last step, comparison based on Scott Knott procedure has been done and concluded that Boehm's approach has done better for COCOMO's datasets.

Z.Toth (2017), have found that IBM's Report Program Generator (RPG)being a high level language system could be explored with the help of Halstead's complexity metrics, which leaves a great impact in metric group. It has been hypothesized that in a system, a disjoint group been formed by Halstead's metrics could be included in the description of warning occurrences at program level [19].Main basis of this hypothesis is to find out the compatibility between the metrics been calculated by analyzing the principal component on 348 RPG programs and 7475 subroutines.

Z.T. Abdulmehdi et. al (2014),have proposed the improvement in the COCOMO II model's accuracy in terms of calculating software efforts based on the alterations in cost drivers and scale factor. The observations are made in the two stage estimation where, estimation variable is multiplied with the constant and then the addition or deduction of some effort can result to the accurate effort estimation [20]. The results calculated in Magnitude of Relative Error (MRE), MMRE, Root Mean Square (RMS) and Relative Root Mean Square (RRMS) were plotted in the graphs.

M.Pauline et. al (2013),have compared the performance of effort estimation methods by calculating effort in KLOC values of various models. They have vastly discussed the calculation of Function Point (FP) approach of the models in detail and concluded that the adjustments required in the FP approach need to be grouped in order to get consistent results. Fuzzy logic being considered as a quality factor, has been included and used as an adjustment factor [21]. As a result of which, occurrence of error gets minimized in the counting process. Hence, by improving adjustments for size of the function methods proves raising the bar to least effort.

2.1. TABLE OF COMPARISON

Table 2: Comparison of Literature Review

S. No	Author's name	Year	Description	Advantages	Disadvantages
1	C. Thirumalai, R. Shridharshan, R. Reynold[15]	2017	Accuracy in Effort estimation using Halstead's metrics and comparing with COCOMO II model.	Actual time is predicted precisely for Python Program as well as for Java program using Halstead's metrics.	The output depending upon Strout's standard value is mainly used to calculate only time precisely.
2	M. Madheswaran, D. Sivakumar[16]	2014	Enhancing the prediction accuracy in COCOMO model with the help of neural network.	COCOMO model gets mapped to neural network and that enhances the predication accuracy using neural network in COCOMO model for software project.	There are some results which prove that neural networks predicts increased effort as compared to actual effort.
3	P. Le, V. Nguyen[17]	2017	A comparison analysis based onapproaches like k-nearest neighbours and COCOMO calibration.	A combination of k-nearest neighbors and COCOMO calibration proving to perform better among other solo approaches.	Works for definite amount of dataset only and is not applicable for new or different data sets.

4	T. Menzies, Y. Yang, G. Mathew, B. Boehm, J. Hihm[18]	2016	Comparing older and newer models of SEE and predicting negative results.	Emphasized on the methods of collecting data instead of approaches used.	No better approach has been suggested in order to overcome negative results.
5	Z. Toth[19]	2017	Defining and developing prototype of Halstead's metrics along with evaluating maintainability index variants of RPG.	Use of High level language and new technology i.e. IBM's RPG.	No other kind of language, used in general like Java, C++ ,Python etc. , are examined.
6	Z.T. Abdulmehdi, M.S.S. Basha, M. Jameel, P. Dhavachelvan[20]	2014	COCOMO II's variant for improved SEE by using cost drives and scale factor.	Have made the previously built model more reliable by implementing changes instead of proposing a new one.	Challenging a well established approach and redefining the variables need more experience and is time consuming as well.
7	M. Pauline, P. Aruna, B. Shadaksharappa[21]	2013	Comparison of former SEE methods and proposing quality factor as an adjustment factor for better estimation using fuzzy logic.	Efforts can be calculated in the early stages of software development using Function points.	It only gives the approximate value of planned effort.

3. CONCLUSION

In this paper, various models of software effort estimation have been considered along with the advantages and disadvantages of the techniques used under different criterion. It is not appropriate to choose the best model among all, as all the models are good in their own ways. Still, we have compared these models based on the work they have done in the last few years. Most popular models, COCOMO and COCOMO II are investigated based on formulae using KLOC values. Same KLOC value formulae with little variation in constants are considered for Halstead's metric for effort estimation and for Baliey-Basili and SEL as well. After the comparison made in this paper, the future work possible after this study, is to learn new effort estimation techniques; which are making it easy to work on software project estimation.

4. REFERENCES

[1] S.Kuan,"Factors on Software Effort Estimation", International Journal of Software Engineering & Applications (IJSEA), Vol.8, No.1, Jan. 2017.

[2] A. Ren, C. Yun, "Research of Software Size Estimation Method", IEEE, vol.18, issue 4, pp- 154-155, 2013.

[3] M.M. Kirmani, "Software Effort Estimation in Early Stages of Software Development: A Review", International Journal of Advanced Research in Computer Science, Vol. 8, No. 5, 2017.

[4] H. Duggal, P.Singh, " Comparative Study of the Performance of M5-Rules Algorithm with Different Algorithms", Journal of Software Engineering and Applications, pp-270-276, April 2012.

[5] W. Rosa and C. Wallshein, "Software Effort Estimation Models for Contract Cost Proposal Evaluation" presented at ICEAA Professional Development & Training Workshop,17, 2017.

[6] P.Sandhu, M.Prashar, P.Bassi, A.Bisht, "A Model for Estimation of Efforts in Development of Software Systems" International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol.3, No.8, 2009 .

[7] S. Sabrjoo, M. Khalili, M. Nazari, "Comparison of the accuracy of effort estimation methods", International Conference on Knowledge- Based Engineering and Innovation,Nov. 2015.

[8] R.W. Selby, *Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management and Research*, New Jersey: Wiley,pp. 133-135, 2007.

[9] A.F. Sheta, S. Aljahdali, "Software Effort Estimation Inspired by COCOMO and FP Models: A Fuzzy Logic Approach", International Journal of Advanced Computer Science and Applications, Vol. 4, No. 11,

2013.

[10] S.Sarwar, M. Gupta, "Proposing Effort Estimation of COCOMO II through Perceptron Learning Rule", International Journal of Computer Applications, Vol. 70, No.1, May 2013.

[11] T.N. Sharma, "Analysis of Software Cost Estimation using COCOMO II", International Journal of Scientific & Engineering Research Vol. 2, Issue 6, June 2011.

[12] S. Chalotra, S. K. Sehra, Y. S. Brar, N. Kaur, "Tuning of COCOMO Model Parameters by using Bee Colony Optimization", Indian Journal of Science and Technology, Vol. 8 No.14, July 2015.

[13] P.S. Espalanda, E.A. Albacea, "Assessing Accuracy of Formal Estimation Models and Development of an Effort Estimation Model for Industry Use", International Journal of Information Technology and Business Management, Vol.7, Nov. 2012.

[14] V.Chandra, "Software Effort Estimation: A Fuzzy Logic Approach", International Journal of Computer Applications, Vol. 103, No.9, October 2014.

[15] C.Thirumalai, R. Shridharshan, R. Reynold, "An Assessment of Halstead and COCOMO Model for effort Estimation", International Conference on Innovations in power and advanced computing technologies, IEEE, 2017.

[16] M.Madheswaran, D.Sivakumar,"Enhancement of prediction accuracy in COCOMO model for software projects using neural network", 5thICCCNT, IEEE, July 2014.

[17] P. Le, V. Nguyen, "A k-nearest neighbors approach for COCOMO calibration",4th NAFOSTED Conference on Information and Computer Science, IEEE, 2017.

[18] T. Menzies, Y. Yang, G. Mathew, B. Boehm, J. Hihm, "Negative results for software effort estimation", Empirical Software Engineering, Springer, Nov. 2016.

[19] Z. Toth, "Applying and evaluating Halstead's complexity metrics and maintainability index for RPG", Springer International Publishing, 2017.

[20] Z.T. Abdulmehdi, M.S.S. Basha, M. Jameel, P. Dhavachelvan, "A variant of COCOMO II for the improved Software Effort Estimation", International Journal of Computer and Electrical Engineering, Vol.6, No. 4, Aug. 2014.

[21] M. Pauline, P. Aruna, B. Shadaksharappa, "Comparison of available methods to Estimate Effort, Performance and Cost with the proposed method", International Journal of Engineering Inventions, Vol.2, No. 9, May. 2013.