Enhancing the Software Effort Prediction Accuracy Using Reduced Number of Cost Estimation Factors with Modified COCOMO II Model

D. Sivakumar[1], K. Janaki[2]

[1]*Associate Professor, Dept. of CSE, ACS College of Engineering, Bengaluru, Karnataka, India*
[2] *Associate Professor, Dept. of CSE, RajaRajeswari College of Engineering, Bengaluru, Karnataka, India*

Abstract - *The empirical estimation method mainly relies on cost drivers in estimating effort and cost of software projects. The cost drivers and the selection of ranges for a particular cost driver will not be same for all models and situations. The variety of cost drivers and its properties in the standard COCOMO II model in view of recent scenario is attained more focus on research interest. The main objective of this work is to analyze the COCOMO II model cost drivers and the impact of some specified cost drivers in estimating effort and cost of software projects. In this paper the ranges of cost drivers and its values are adjusted according to the recent industrial situations and needs. The number of cost drivers is reduced to 13 and the efforts are estimated using this newly modified cost drivers. This model proved its improved efficiency in estimation with reduction in percentage of MRE and MMR values.*

***Key Words*: Effort Estimation, Software Project, Cost Drivers, Modified COCOMO II, MRE, MMRE.**

1. INTRODUCTION

Estimations are indispensable in software projects to support the decision building in different phases Boehm [1]. The very first decision on a project is evaluating, in which it is acknowledged that, whether the project is usually and economically feasible or not Boehm et al., BW & Valerdi, R [2][3]. The effort required to make the software is a vital factor in building decision, as software projects seldom comprise major cost items other than salaries and interrelated side expenses. Even before starting a project, there could be deliberate forecast activities to find out the potential relevance domains and projects Charette & Chen et al., [4][5].

Estimating the software project development effort in early development is a tedious job for the software project managers in the current industrial situations De Jong [6]. In this paper it is aimed to analyze and identify the changes required in the already developed COCOMO II post architecture model and the cost drivers are classified and reframed according to the recent industrial scenarios Denver et al., & Elyassami et al., [7][8].

The COCOMO II post architecture model has 17 effort multipliers whereas the early design model uses only six effort multipliers which are also called as cost drivers Fischman [9]. The COCOMO II early design model is a simplification of the post architectural model Galorath & Evans [10]. All type of COCOMO model analysis is made based on the impact of each software cost attribute in estimation and in specific development situations Cuadrado et al.,[11]. These types of estimation and analysis will helps us to suggest some useful guidelines to the software project managers for better software cost and effort estimation and to maintain the cost of a software project in specified limit for better decision making at different levels.

2. LITERATURE SURVEY

Samson [12] connected neural system model, Cerebellar Model Arithmetic Computer (CMAC) to the expectation of exertion from programming code size. CMAC is an observation and capacity estimate created by Goldberg & Hale [13] [14]. This neural system was prepared for Boehm's COCOMO information set with a specific end goal to foresee exertion from size, in the same way relapse procedures were connected for expectation purposes.

Point by point audit of distinctive studies on the product development effort was given by Jorgensen [15] with the principle objective of contributing and supporting the master estimation research. Neural systems have the learning capacity and are great at displaying complex nonlinear relationships gives more adaptability to incorporate master information into the model.

The Standish exploration, gathering said in the CHAOS report uncovers the significant crisis joined with the fate of the product ventures. Jorgensen et al [16]. This likewise demonstrates that the expense overwhelm connected with it is 189%. A dominant part of investigation on utilizing the neural systems for programming expense estimation, are centered around demonstrating the COCOMO strategy, for instance, in Attarzadeh et al [17] a neural system has been proposed for estimation of programming expense.

Recardo de Aroujo et al [18] exhibited a cross breed savvy model to plan the morphological rank linear perceptrons to take care of the product cost estimation issue by utilizing the modified genetic algorithm with a gradient descent strategy to upgrade the model. Shepperd & Schofield [19] depicted an option way to deal with the exertion estimation in view of the utilization of analogies some of the time alluded to as case based thinking. The principal guideline is to portray ventures as far as elements, that is, the quantity of interfaces, the advanced technique or the measure of the useful necessities Goldberg & Hall [20][21].

3. COST DRIVER SCENARIOS

The range of effort multipliers are adjusted or modified based on the changes required for present industrial situations. The reusability, risk management and quality of the software product are considered as a key goal in changing the cost drivers and its ranges Helm & Idri [22] [23]. The rating levels are assigned or altered based on the impact in recent development activities like low, very low, high, very high, extra high and nominal. Software project effort multipliers also called as cost drivers of COCOMO II post architecture model are categorized into four major areas viz.,

1. Product factors
2. Platform factors
3. Personal factors
4. Project factors

Product factors

Four cost factors are grouped in this category and these factors or cost drivers will decide the characteristics and the cost of the software project based on its impact of the effort required to develop the software project. The ranges selected for the cost drivers will decide the complexity levels and the effort required for developing software project.

Platform Factors

These factors are measured based on the impact of target machine hardware and software infrastructure in developing the software project. It consists of three factors in which the effort required to develop the code that is to be executed in a target machine's hardware and software platforms.

Personal Factors

The personal factor is used to measure the capacity of the individual person involved in developing the software product. It is considered as the major influencing factor, because the people who develop the software product must have a very good capability level in different aspects to produce better software product.

Project Factor

The software development technology, the environment in which the software is to be developed and changes in the development schedule are taken into account with the help of three different project factors.

4. SENSITIVITY OF COST DRIVERS IN EFFORT ESTIMATION

Almost all empirical effort and cost estimation models estimates its output by taking the major input factor as the cost drivers and the scale factors Chiu et al., [24]. These models reveal the problem of instability in the values of the cost drivers and the scale factors that affects the sensitivity of effort. Almost of the models encompasses one or more inputs for which a little change input will result in huge change in project effort and schedule Iman Attarzadeh [25]. Based on literature study, analysis and in view of analogy made for this research study it is considered that the personal factor and the product factors are the sensitive input in determining the effort of a software project.

In this study the factors CPLX, ACAP, PCAP are considered as the sensitive input to the model. In addition to the sensitivity of the cost factor, ACAP factor is considered with the experience of the analyst. The programmer capability factor is taken as an addition of the platform experience, and application experience factors. Based on these conceptual ideas the new set of cost drivers that is the effort multipliers framed are listed in Table 1.

Table -1: Modified new set of cost driver

S.No	COST ATTRIBUTES	DESCRIPTION
PRODUCT FACTORS		
1.	RELY	Required Software Reliability
2.	DATA	Database Size
3.	CPLX	Product Complexity
4.	DOCU	Documentation and Reusability
PLATFORM FACTORS		
5.	TIME	Execution Time
6.	STOR	Main Storage Constraint
7.	PVOL	Platform Volatility
PERSONAL FACTORS		
8.	ACAP	Analyst Capability
9.	PCAP	Programmer Capability
10.	PCON	Personal Continuity
PROJECT FACTORS		
11.	TOOL	Software and Language Tool Experience
12.	SITE	Multisite Development
13.	SCED	Required Development Schedule

The new set of cost drivers has 13 effort multipliers and it is framed by considering the individual experience and capability, which are important for the better team's ability.

The definition of the modified COCOMO II model with the new set of attributes had the following form.

$$Effort = AS^{1.01+\sum_{i=0}^{5} SF_i} \prod_{i=1}^{13} EM_i \quad (1)$$

Where,

A - Multiplicative constant
S - Software project size in KSLOC
SF - Scale Factors
EM - Effort Multipliers

The efforts of the modified COCOMO II model with the new set of 13 effort multipliers are estimated using the

Equation (1). The mean square error (MRE) and the magnitude of mean square error are calculated using the equation (2) and (3).

$$MRE = \frac{|AE - EE|}{AE} \times 100 \quad (2)$$

$$MMRE = \frac{\sum_{i=1}^{N} MRE}{N} \quad (3)$$

Where,
AE- Actual Effort and EE – is the Estimated Effort
'N' - is the total number of projects considered for evaluation

5. RESULTS AND DISCUSSIONS

The Table 2 provides the effort estimated and the percentage magnitude of relative error by the modified COCOMO II model with a new set of EM. When compared to the COCOMO II model effort, the modified COCOMO II with new EM model's estimated effort is highly closer to the actual effort. The project with project ID 5 is the best example for proving the performance of the model.

Table -2: Effort and %MRE estimated with New EM

S.No	Project ID	ACTUAL EFFORT	COCOMO II EFFORT	Modified COCOMO II with New EM EFFORT	%MRE of COCOMO II	%MRE of Modified COCOMO II with New EM
1.	1	2040	2018	2037	1.08	0.15
2.	5	33	39	34	18.18	3.03
3.	9	423	397	420	6.15	0.71
4.	11	218	190	212	12.84	2.75
5.	26	387	391	385	1.03	0.52
6.	34	230	201	227	12.61	1.3
7.	42	45	46	43	2.22	4.44
8.	47	36	33	34	8.33	5.56
9.	50	176	193	171	9.66	2.84
10.	51	122	114	120	6.56	1.64
11.	54	20	24	18	20	10
12.	56	958	537	955	43.95	0.31
13.	61	50	47	47	6	6
MMRE					11.43	3.01

In view of the magnitude of relative error the modified COCOMO II model with new EM generates very les MRE. The mean magnitude of relative error (MMRE) of the COCOMO II model is 11.43 whereas the MMRE of modified COCOMO II with new EM is 3.01.

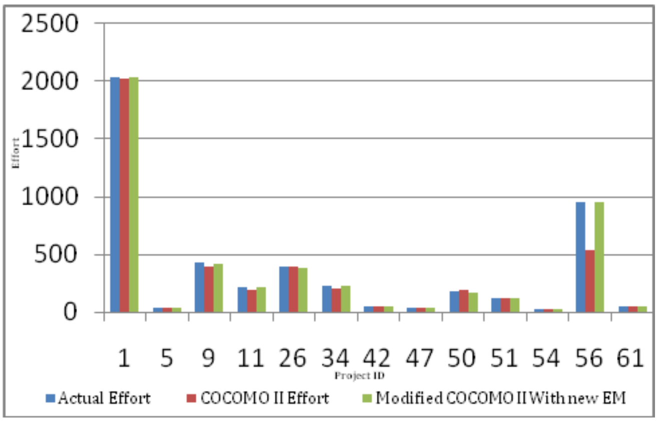

Chart -1: Effort estimated with new EM

Chart 1 shows the pictorial representation of the effort estimated using the modified COCOMO II. From this figure it is clearly understand that the estimated effort of the modified COCOMO II with a new set of EM is very closer to the actual effort.

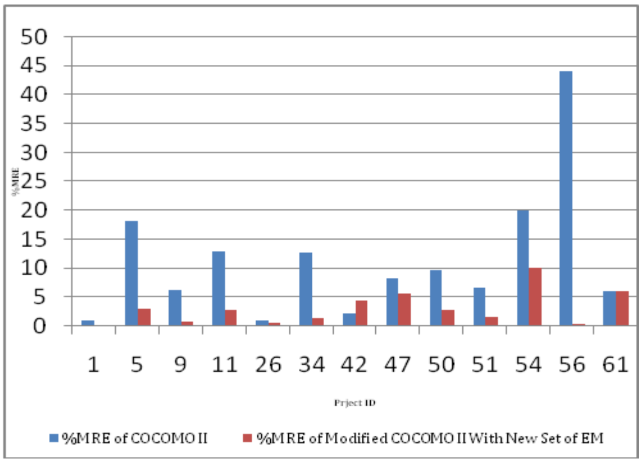

Chart -2: Effort estimated with new EM

Chart 2 shows the percentage magnitude of relative error generated by the modified COCOMO II model with a new set of EM. This graphical representation shows that almost all the %MRE values are comparatively lesser than the COCOMO II model % MRE even though the project with project ID 61 generates the %MRE is equal to the %MRE of COCOMO II model.

6. CONCLUSIONS

The experimental study based on the modified COCOMO II with new EM model is proved with its ability in estimating the effort for the different software projects. This study also suggests some of the findings to the software project managers based on the sensitivity analysis of cost factors.

The various estimations made in this study was identified that some of the cost drivers are highly sensitive. So at most care must be taken in determining such factors. This analysis may be further extended to identity the time of development, fitness estimation and etc. In addition this work may be extended to fit into the framework based development environments.

REFERENCES

[1] Boehm 1981, 'Software Engineering Economics', Prentice Hall.

[2] Boehm, BW & Valerdi, R 2008, 'Achievements and challenges in cocomo-based software resource estimation', IEEE Software, vol. 25, no. 5, pp. 74-83.doi:10.1109/MS.

[3] Boehm, BW, Horowitz, E, Madachy, R, Reifer, D, Clark, BK, Steece, B, Brown, AW, Chulani, S & Abts, C 2000, 'Software cost estimation with COCOMO II', Prentice Hall.

[4] Charette, RN 2005, 'Why Software Fails [Software Failure]', IEEE Spectrum, vol. 32, no. 9, pp. 42-49.

[5] Chen, Z, Menzies, T, Port, D & Boehm, B 2005, 'Finding the right data for software cost modeling', IEEE Software, vol. 22, no. 6, pp.38-46.

[6] De Jong, K 1992, 'Are genetic algorithms function optimizers?', Proc. Sec. Parallel Problem Solving From NatureConference, pp. 3-14, The Netherlands: Elsevier Science Press.

[7] Denver, CO, Martin-Marietta, Ratliff & Robert, W 1993, 'SASET 3.0 user Guide' Oruada, Gerald L, 'Software Cost Estimation Models: A Callibration, Evaluation, and Compartison', Air force institute of Technology.

[8] Elyassami, S & Idri, A 2012, 'Investigating effort prediction of software projects on the ISBSG dataset', International Journal of Artificial Intelligence & Applications (IJAIA), vol. 3, no. 2, pp. 121-132.

[9] Fischman, L, McRitchie, M & Galorath, DD 2005, 'Inside SEER-SEM', CrossTalk, The Journal of Defense Software Engineering, vol. 18, no. 04, pp. 26-28.

[10] Galorath, D & Evans, MW 2006, 'Software sizing, estimation, and risk management: When performance is measured performance improves', Boca Raton, FL: Auerbach Publications.

[11] Cuadrado-Gallego, JJ & Rodri 2010, 'Analogies and Differences between Machine Learning and Expert Based Software Project Effort Estimation', 11th ACIS International Conference on Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD).

[12] Samson, B 1997, 'Software cost estimation using an albus perceptron', journal of Info & Softw., vol. 39, pp. 55-60.

[13] Goldberg, D 1989, 'Genetic Algorithms in Search, Optimization and Machine Learning', New York, Addison-Wesley publisher.

[14] Hale, J, Parrish, A, Dixon, B & Smith, RK 2000, 'Enhancing the Cocomo estimation models', IEEE Software, vol. 17, no. 6, pp. 45-49.

[15] Jørgensen, M 2004, 'A Review of Studies on Expert Estimation of Software Development Effort', Journal of Systems and Software, vol. 70, pp. 37-60.

[16] Jørgensen, M 2005, 'Practical guidelines for expert-judgment-based software effort estimation', IEEE Software, vol. 22, no. 3, pp. 57-63. doi:10.1109/MS. 73.

[17] Attarzadeh, I & Siew Hock Ow 2010, 'Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks', IEEE International Conference on Computer Engineering and Technology (ICCET), vol. 3, pp. V3-487 - V3-491.

[18] Ricardo de A Araujo, Adriano LI de Oliveira & Sergio Soares 2012, 'Hybrid Intelligent Design of Morphological Rank Linear Perceptrons for Software Development Cost Estimation', 22nd International Conference on tools with Artificial Intelligence.

[19] Shepper, M & Schofield, C 1991, 'Estimating software project effort using analogies,' IEEE Tran. Software Engineering, vol. 23,pp. 736-743.

[20] Goldberg, D 1989, 'Genetic Algorithms in Search, Optimization and Machine Learning', New York, Addison-Wesley publisher.

[21] Hall, M & Holmes, G 2003, 'Benchmarking attribute selection techniques for discrete class data mining', IEEE Transactions On Knowledge And Data Engineering, vol. 15, no. 6, pp. 1437-1447.

[22] Helm, JE 1992, 'The viability of using COCOMO in the special application software bidding and estimating process', IEEE Transactions on Engineering Management, vol. 39, no. 1, pp. 42-58.

[23] Idri, A & Mbarki, S 2004, 'Validating and understanding software cost estimation models based on neural networks'. Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. International Conference on Information and Communication Technologies, pp. 432-438.

[24] Chiu, NH & Huang, SJ 2007, 'The adjusted analogy-based software effort estimation based on similarity distances', Journal of Systems and Software vol. 80, no. 4, pp. 628-640.

[25] Iman Attarzadeh & Siew Hock Ow 2010, 'Soft Computing Approach for Software Cost Estimation', Int.J. of Software Engineering, IJSE vol. 3, no. 1, pp. 3-12.