A Comparative Study of Effort Estimation Models in Software Engineering - A Review

Sarbjeet Kaur, Navdeep Sethi

Abstract

Accurate estimation of the cost of software projects is one of the most desired capabilities in software development organizations. Precise cost estimates not only help the customer make successful investments but also assist the software project manager in coming up with appropriate plans for the project and making reasonable decisions during the project execution. An estimate is not really a prediction, it is a management goal. Assessment of the proposed system consists of evaluating whether the required functionality can be achieved with current affordable technologies. Effort and time required to develop software can be computed by estimating the project size. Parametric models use effort drivers representing characteristics of the target system and the implementation environment to predict effort. In Parametric model, there is a use of mathematical expressions for estimating cost associated with software development. Inaccurate cost estimation may lead to project failure, huge overruns and performance compromises as a consequence. The proposal is to sensibly analyze hybrid parametric models with size and cost estimation models to allow the authors to determine a set of homogeneous projects.

Index Terms– COCOMO, effort estimation, parametric estimation model, Function Point, Hybrid model.

I. INTRODUCTION

Software engineering is all about the yielding of software, from its commencing stage to its final stage. Computer science is relevant to the software engineering in numerous demeanors. The nature and the complexity of the software system are changing. Software engineering is about evolving, domicile, and sustains software systems. Software engineering is a set of problem solving skills, expertise, methods and techniques applied to develop and yield useful systems that resolve many problems like practical problems. Software engineer is crucial to hold software engineering projects which find out, accomplished software and tells its performance. A control about their work using some techniques and tools lean on the resources presented and problem to be solved.

II. Software Cost Estimation

The Software Cost Estimation is concern for software development adept, experts and managers of software systems. Software cost estimation plays a requisite ingénue in software engineering as the success or failure of project exclusively depends on it. Cost estimation's deliverables like staff requirements, schedule and effort are essential chunk of information execution of project. It accommodates inputs for project scenario, project planning, control, aggregate, and progress monitoring & staff allocation. Illogical and ambiguous estimates are the mainly cause of project failure. So, the capability of the system to find out correct time and cost of the software is clamored for the progress of the system. It is challenging ingredients for software industry to conceive the explicit estimation of software development. Success in managing projects is a critical factor for the success of the whole organization. The

ability of the system to find out accurate time and cost of the software is very necessary for the success of the system. So there are different way and different types of methods to estimate cost and time of the software.

Models for Cost Estimation:

1. COCOMO
2. Use Case Point Estimation
3. Function Point Based Estimation
4. Expert Judgment

COCOMO

It is Constructive Cost Model. It is very effective and oldest model for cost estimation **[9] [4]**. It is autonomous model which is well documented and cannot be depended upon any software vendor. In COCOMO model line of code is predicted. In this model we can understand the complexity of the system because of its observance essence. This model is constructed to evaluate the cost estimation of the software development.

There are three levels in COCOMO model:

1. Basic COCOMO:

It computes software development cost and effort as a function of program size. It is static and single valued model [9]. [3]There are three modes within Basic COCOMO:

a) Organic Mode
b) Semidetached Mode
c) Embedded Mode

Table.1 Modes of COCOMO

Project type	A	B
Organic (simple in terms of size and complexity	3.2	1.05
Semidetached(average in terms of size and complexity)	3.0	1.12
Embedded(Complex)	2.8	1.20

2. Intermediate COCOMO:

It is an accession to the basic model that computes software development effort by adding a set of cost drivers. [10] In this 15 cost drivers are used to find out cost estimation of projects graded from very low to very high.

3. Detailed COCOMO

It is an extension of intermediate COCOMO [9]. In this cost driver is added to effort multiplier at each phase to calculate the cost drivers. It uses different multiplier for each cost attribute. COCOMO 1 is also known as COCOMO 81.The basic COCOMO formula for effort estimation is:

Effort=a*(kloc) ^b
Values of a and b depends upon the type of projects.

COCOMO 2

This model simplified the cost estimation of COCOMO81 by reducing number of parameters. The parameters which are used in COCOMO 2 are totally different from its typical value. [12] COCOMO II has the advantage of supporting lines of code and function points as input simultaneously. It also has a simple estimation process and transparent algorithm. COCOMO 2 models have two types of parameters set. First is external set and it can be equivalent to matrix glimpse loosely. The thesaurus of the model can be used easily while dealing with stakeholders. The second set is internal which is used in different purpose than first set. There are many tools which are available in market which calculated the appropriate results for cost estimation [13].

COCOMO 2 model conserves the originality of COCOMO model i.e. openness of the COCOMO. There are three stages which are accessible in COCOMO 2. First stage is pursuing the prototyping model with the help of application model capability. The next phases normally attend investigation of architectural alternatives or incremental development strategies. When project is ready to develop then it should have life- cycle architecture.

[9]The basic differences between both models are as follow:

Table.2 Differences of COCOMO 1 and COCOMO 2

COCOMO1	COCOMO2
It is basic model.	It is extension model of basic model
It follows waterfall model	It follows three phases concept.
There are 15 cost drivers are present in COCOMO1.	There are 17 drivers present in this model
There are data 63 data points in it	There are 61data points in it.
In this model reengineering concept is followed.	Software reusability is used in it
It is measured in KDSI	It is measured in KLOC

Limitations of COCOMO

There are some limitations of this model which are as follows:

1. COCOMO starts estimation from the design phase and continues till the end of integration phase of cost and schedule of the project. A separate estimation model should be used for remaining phase [9].
2. KDSI cannot measure the size.

3. It is difficult to exactly estimate in this when maximum estimations are required.
3. A new estimation may show over budget or under budget for the project when to revise the cost of the project. This may lead to a partial development of the system [9].

Use Case Point

A functional ambit of the system is defined by the use case point. It assists as a top-down approach. It is well suited for project approximately and planning. This estimation method counts the number of transactions in each use case. A transaction is an event occurring between the system and an actor .The Use Case Points (UCP) method accommodate the ability to estimate the man hours a software project desire from its use cases. [3] It analyzes the use case actors, scenarios, and various technical and environmental factors and abstracts them into an equation. The UCP equation is composed of three variables:
1. Unadjusted Use Case Points (UUCP).
2. The Technical Complexity Factor (TCF).
3. The Environment Complexity Factor (ECF)

Function Point Based Estimation

FPA is an ISO organized which is used to find out the functional size of the system. The functional size indicates the lot of functionality which is necessary. It is not dependent upon any technology which used to implement the system. FPA express the size of the information in a number of function units. So its measurement unit is function units.[12] Function point method decomposes the complex system into simple subsystems based on software documentation.[4]Value adjustment factor is another important factor for function point analysis which is calculated by taking the sum of 14 general system characteristic
FPA=VAF*UFP
FPA=function point analysis
VAF =value adjustment factor
UFP=unadjusted function point

Expert Judgment based Methods: Expert judgment is often based on management or estimator recollection of past projects that may or may not have been documented [13]. There is no need of historical data for this method. prosperity of using expert judgment is that the estimation is customized to the specific organizational culture, hence this estimation technique is supplementary accurate rather than general Algorithmic approaches.[13] Expert judgment is a non-structured process even though in many cases it has been proven to give a better accuracy than using other techniques. The final estimation of the experts is subjective and based on feelings and logic

The estimation based on multiple factors is discussed in the following paper*:*

A. Accuracy of Contemporary Parametric Software Estimation Models

This paper [1] shows the accuracy of parametric software estimation models. In their paper, they compared four parametric software estimation models in term of their effort and duration prediction accuracy. 51 real project data are used to analyze abilities of the models which compare with the actual effort and duration values. The results of the models that are

investigated are par on accuracy. The future work suggested by them can be considered as incorporating historical data for adjustment purpose to have more insight into capabilities and strength of these methods and tools.

B. Efforts estimation by combining the use case point and COCOMO

This paper [3] combines the Use Case point and COCOMO. They forecast the Line of Code with the help of Use Cases. Explained by them that Use Case used in the method must be more specific not more generalized .The Use Cases gained wide popularity in software effort estimation. Results obtained using use cases are widely applicable. A strong monitoring policy is always required to make estimation as a success. They have to make a check list with the date of completion and must follow the checklist. If work is not done on the time some necessary action must be taken to compensate the deviation. To estimate the KLOC divide the project into module and module into the sub module until we are able to estimate the KLOC. Use Case Point shows the functional requirement of the system .So it is one of the good way of estimation. They also tried to illustrate that how we can combine Use Case and KLOC.

C. *To Enhance the Effort Estimation Accuracy of COCOMO Model using Function Point*.

This paper [4] combination of COCOMO model and Function point to produce less effort than COCOMO. Estimating efforts accurately determines whether the development of software is failure or success. As stated in paper that among all the models effort provided by the COCOMO are close to actual effort. In their paper, they are proposing a hybrid model of COCOMO and function point that produces estimated efforts less than COCOMO and function point alone

D. Predictive and Stochastic Approach for Software Effort Estimation

This paper [6] proposed Particle Swarm Optimization technique is proposed which drive on data sets which are clustered using the K-means clustering algorithm. PSO has been employed to generate parameters of the COCOMO model for each cluster of data values. Back Propagation technique is used to train the Neural Network. COCOMO 81dataset is used for testing and also the results have been compared with standard COCOMO model and as well as the neuron fuzzy model. It is concluded from the results that the neural networks with efficient tuning of parameters by PSO operating on clusters, can generate better results and hence it can function efficiently on ever larger data sets.

E. Proposing an Enhanced Artificial Neural Network Prediction Model to Improve the Accuracy in Software Efficient

The following paper [5] describes, in software development, the project manager has to face the problems regarding cost, time and staff estimation. This is one of the critical tasks in software development process. This paper provides better view oh hybrid model ANN-COCOMO i.e. COCOMO model using artificial neural network for effective effort estimation. Software estimation is classified as algorithmic and non algorithmic technique. COCOMO is considered as the best model that follows algorithmic techniques such as regression technique that is based on historical data. ANN is a mathematical technique to calculate the working condition of human brain.

ANN basically makes fine adjustments of attributes using historical data. Due to change in the business environment, the relationship among attributes becomes vague. To overcome this problem, this paper proposed ANN-COCOMO model.

F. Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimization.

The paper [7] is a collaboration of artificial neural network (ANN) and Constructive Cost Model (COCOMO), which expanded by Particle Swarm Optimization (PSO). PSO-ANN-COCOMO II model accurately estimate the cost of Software development. This revised model not only raised the speed of artificial neural network but also resolve the problem of dependency of initial weight in learning ability of artificial neural network. With keeping the advantages of COCOMO model, this model get better the learning capability of original model.

F. Improvement in COCOMO Model for Effort Estimation using Neural Networks

The paper [10] The COCOMO model is the model which is used to estimate the efforts of the projects. The improvement in the COCOMO model will be proposed which will be based on function point and neural networks. The function point divide whole project into individual functions and neural networks is used to calculate KLOC of each function. In this paper, enhancement in COCOMO model has been done using Boltzmann Learning

G. A Survey on Software Effort Estimation Techniques

This paper [2] presents techniques and models of effort estimation. Comparison among several approaches is being done and the technique that produces the most accurate result serves as a measure of selection. They specified in there paper that every technique has its own merits and demerits. They suggested in their paper that there is no single technique that can run away from all the shortcomings and can be globally accepted, so the future work suggested in her paper is hybridization of several approaches as an alternative to produce realistic estimates.

H. A Bayesian Network Model of the Particle Swarm Optimization for Software Effort Estimation

This paper [8] presents the new hybrid Bayesian Network model of PSO for effort estimation. It proved that Bayesian Network with PSO gives more accurate results than other existing techniques. Compare the proposed model with COCOMO and Bayesian Regulation Neural Network Model and it is found that the developed model provides better estimation.

III. COMPARISON

In paper [1] the comparison is made between algorithmic models i.e., COCOMO-II, SEER-SEM, SLIM and true planning and reduced error rate to provide better estimation. In paper [3] a hybrid model of COCOMO and Use case point is being evaluated and provide good alternative. In paper [4] the combination of function point and COCOMO will produce fewer efforts than COCOMO or any other model. A hybrid model of COCOMO and function point is designed that produces estimated efforts less than COCOMO and function point alone. In paper [5] a hybrid model of ANN-COCOMO II is evaluated. It provides more

accurate by 17.1% when compared to the COCOMO II Model In paper [6] Evaluate PSO with K-Mean clustering algorithm and results compared with standard COCOMO Model and neuron fuzzy model and can perform efficiently on ever large data sets. In paper [7] like a hybrid model ANN-COCOMO II- PSO perform better results than when ANNs or COCOMO are used in isolation. PSO-ANN-COCOMO II has a progress of 3.27% in software effort estimation precision than the original artificial neural network Constructive Cost Model (ANN-COCOMO II). In paper [8] Active the Bayesian Network with PSO gives more accurate results than other existing techniques. Evaluation performed on NASA 93 datasets to verify the model and also compare the proposed model with COCOMO and Bayesian Neural Network Model and it got results that the combined model gives better estimation. In paper [9] COCOMO model, SLIM model and hybrid models are compared in terms of MRE value. The comparison shows that higher MRE value is of SLIM model, the second higher is of COCOMO model and third higher value of MRE is of COCOMO+FUNCTION POINT model. SLIM modal is also the cost estimation modal which works on KLOC and cost driver values. It has higher error rate as compared to other models like COCOMO & Function Point. In paper [10] in this function point and neural networks are being evaluated to enhance the performance of COCOMO model. The function point divide whole project into individual Functions and neural networks are used to calculate KLOC (lines of code) of each function. It improves the performance of COCOMO model by using Boltzmann Learning. Developed model shows improve MRE value as compared to the existing COCOMO Model. In paper [11] a hybrid model based on PSO and DE algorithms to provide more detailed and efficient estimate. The hybrid model works well with the incomplete and ambiguous input data and it can operate reliably in SCE. The results showed that the accuracy of PRED in the hybrid model has been increased by about 1.34 times. The accuracy of the hybrid model is more that show its accuracy is 1.87 times higher than COCOMO model.

The comparison of the above studied estimation techniques has been done in the following table:

Table.3 Study Of the various effort estimation technique

Model and Technique name	**Dataset**	**Performance Evaluation**
Cocomo-2,Seer-Sem,SLIM,True planning	ISBSG Repository	Provide better estimation than cocomo-2
PSO-ANN-COCOMO 2	COCOMO1 AND NASA93	Powerful tool

Hybrid model of COCOMO and use case point	Successful projects of a company	It gives a better substitutes for software effort estimation
Hybrid model of ANN-COCOMO2	COCOMO1 And COCOMO2	It leads to the betterment in accuracy
PSO with K-Mean	COCOMO81	Faster
Hybrid model of-COCOMO and function point	NASA	Enhanced accuracy
Bayesian network with PSO	NASA 93	Better performance
PSO with Chaos Model	NASA 60	Enhanced precisions by reduced errors
Hybrid of COCOMO and SLIM	NASA	Reduced MRE
Hybrid of COCOMO , function point and neural network	NASA 93	Better results

IV. CONCLUSION

Software development effort estimation is the process of amount of effort required to develop or maintain software based on incomplete, uncertain and noisy input. Software estimation is the mechanism of predicting cost, effort and duration that are required to develop

software. Estimator often depends on number of pragmatism to generate software estimations. Estimating efforts accurately decides the software failure or success. Various models and techniques are applied to estimate the efforts accurately [4]. Exceeded budgets, function that are not developed completely, low quality and partial completion of the project are some of the major factors that results in underestimation or overestimation of the software cost[9]. Estimation of cost and size of the software project are the biggest challenge. Effort estimates may be used as input to project plans, budgets, investment analyses, pricing processes and bidding rounds. To hybrid the parametric models are expected to deliver accurate &failure free estimates and that to within a specified period of time. [14]There is no single technique which is sufficient to do away with all the shortcomings so hybridization of more than one technique and methods can give us more accurate estimate that can be helpful to avoid over-estimation or under-estimation of effort

REFERENCES

[1] Derya Toka, Oktay Turetkan, "Accuracy of Contemporary Parametric Software Estimation Models", ©2013 IEEE.

[2] Himani Rastogi, Misha Kakkar,"A Survey on Software Effort Estimation Techniques", ©2014 IEEE.

[3] Chetan Nagar and Anurag Dixit (2012), "Efforts estimation by combining the use case point and COCOMO", IJCA journal .

[4] Inderpreet kaur walia and Manpreet kaur, "To Enhance the Effort Estimation Accuracy of COCOMO Model using Function Point, IJESRT, December 2013

[5] Iman Attarzadeh, Amin Mehranzadeh, Ali Barati (2012) "Proposing an Enhanced Artificial Neural Network Prediction Model to Improve the Accuracy in Software Effort Estimation", Proc of Institute of electrical and electronics engineers, pp. 487-491.

[6] Rao, Srinivasa, C. H. Hari, and Prasad Reddy PVGD. "Predictive and Stochastic Approach for Software Effort Estimation." Int. J. of Software Engineering, IJSE 6.1(2013)

[7] Zhang. "Improving the accuracy in software effort estimation: Using: artificial neural network model based on particle swarm optimization." Service Operations and Logistics, and Informatics (SOLI), 2013 IEEE International Conference on. IEEE, 2013.

[8] Germanjit Singh Sandhu and , Dalwinder Singh Salaria "A Bayesian Network Model of the Particle Swarm Optimization for Software Effort Estimation," International Journal of Computer Applications (0975-8887)Volume 96–No.4, June 2014.

[9] Tajinder Kaur and Jaspreet Singh, "A Hybrid Model for the Enhancement in Software Effort Estimation ," International Journal of Scientific &Engineering Research, Volume 6, Issue 7, July-2015 ISSN 2229-5518

[10] Priya Verma and Ms Aayushi, "Improvement in COCOMO Model for Effort Estimation using Neural Networks," IJACRN , Volume 4, Issue 5, May, 2016 ISSN: 2278-0658.

[11] Majid Ahadi and Ahmad Jafarian, "A NEW HYBRID FOR SOFTWARE COST ESTIMATION USING PARTICLE SWARM OPTIMIZATION AND DIFFERENTIAL EVOLUTION ALGORITHMS," Informatics Engineering, an International Journal (IEIJ), Vol.4, No.1, March 2016.

[12] Aihua Ren, Chen Yun, "Research of Software Size Estimation Method", ©2013 IEEE.

[13] Poonam Pandey, "Analysis of the Techniques for Software Cost Estimation", ©2013 IEEE

[14] Adanma C. Eberendu, "Software Project Cost Estimation: Issues, Problems and Possible Solutions", International Journal of Engineering Science Invention,June, 2014

[15] Mohammad Azeeh, Software Cost Estimation Based on Use Case Points for Global Software Development", CSIT, ©2013 IEEE