Volume 4 Issue IV, April 2016
ISSN: 2321-9653

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Literature Survey on Various Software Cost Estimation Models

Md.Akram [#1], Devi Prasad Reddy G [#2], Ravi Teja Sriramaneni [#3],
#School of Computer Science and Engineering, VIT University, Vellore.

***Abstract:** The main challenge in the development of large and complex projects is the cost estimation with more accuracy. Many estimation models are introduced in the course of time, which concludes that software cost estimation is not precise and new methods or models should be proposed very often. A detailed overview of existing software cost estimation techniques or models are given by this model. The models are majorly classified in two type's algorithmic and non-algorithmic models. Key factor in the development of new software is the selection of the suitable cost estimation model and it also depicts the strengths and weakness of various cost estimation models. The main objective is to provide a comparative literature analysis of various cost estimation methods or techniques in this paper.*

***Keywords:** analogy; neural networks; putnam's model; Seer-Sem model; software cost estimation.*

I. INTRODUCTION

Nowadays in the recent years , the cost for the of the software end products are expensive in nature , there is elevation in competition .Certain Software's are usually complicated or complex and also produced in bulk or large scale , therefore the software cost must be initially estimated by using the various cost estimation models before spending huge amount of money . There are many reasons for performing cost estimation . Underestimating the cost would result in the poor quality of the product , failure to complete the product in time , extra budget to be approved by the management of the approving system . Overestimating the cost of the product may lead to waste of resources , resulting in losing during the contract bidding .
The software cost estimation can be explained in the following steps as follows.

A. Estimate the size of the development product . Lines of Code (LOC) or Function Points (FP) are generally used for estimating the size as units , but there are other possible measurements of units . A analysis of the pros & cons of each model is discussed in some of the material referenced at the end of this report.
B. Estimating the effort in person-hours or person-months.
C. Estimating the schedule in calendar months.
D. Estimating the project cost (in terms of currency)

This paper gives an overview of various cost estimating models and strengths and weakness of the model which is necessary to understand to estimate a real time software project.

Software cost estimation is mainly categorized or classified in two types as follows: *Algorithmic models*, *Non-algorithmic models*

Mathematical Formula is created or developed combining the various cost and product factors in the algorithmic models where as in the Non-algorithmic model there is no such formula in the estimation of the cost , it is based upon the neural network , fuzzy logic , and the evolutionary computation. This paper deal with both the models algorithmic and the non-algorithmic models.

II. BACKGROUND

Software project failure has been a challenge to the makers of the software. Approximately about 35% of software projects has reported to have failed according to a survey . The reasons for the failure of the software projects have been researched and among the found results or reasons , poor planning of the software projects , unsufficient engineering requirements , unexpected decisions made in the beginning stages of the project and imprecise estimation is the significant factor of failure of the most of the software projects . The software projects conditions are not stable and its state is continuosly dangling therefore severael methods must be presented for the estimation of the project ,each method is suitable for a special project.

III. LITERATURE SURVEY

A. Algorithmic models

The Cost is determined or estimated as a mathematical function of product, project and process. Project managers estimate the attribute values.

$$\text{Effort} = a * \text{Size}^{b} * m$$

a = organization-dependent constant,
b indicates the disproportionate effort for larger projects and
m = multiplier reflecting product, people and process attributes
Most of the models are similar only the values of the a , b , m change .
Each algorithmic model uses an equation to estimate:

$$\text{Effort} = f(x_{1}, x_{2}, x_{3}, \ldots\ldots x_{n}).$$

Where $x_{1}, x_{2}, x_{3}, \ldots\ldots x_{n}$ denote the cost factors.
The All cost factors using in these models are: Where D= Relevant to the staffing aspects
Se= Effective size found in the previous equation.
Cte= effective technology.
Once effort is obtained, duration is solved using the following equation
Product factors: Reliability, complexity of the project, size used by database , reusability , documentation matching to life-cycle needs.
Computer factors: execution time constraint, computer turnaround constraints, main storage constraint, volatility of platform.
Personal factors: experience in application , capability in programming , experience in platform , tool and language experience.
Project factors: Use of software tools, required schedule development .[15]
Advantages of algorithmic model are as follows:

1) Possibility of repeatable estimation.
2) Simple modification of input data, refine and customize formulas. [15]

Disadvantage of algorithmic model are its poor sizing inputs and inaccurate cost driver rating will result in inaccurate estimation [15].
List of algorithmic models explained in this paper are

a) *SEER-SEM (Software Evaluation and Estimation of Resources-Software Estimation Model)* : The SEER-SEM model was proposed by Galorath Inc(Galorath,2006) in 1980 . Most parameters in this method are business and commercial projects usually use SEER-SEM as their primary method of estimation . Size of the software is the key feature in this method and also effect size parameter .

Se is computed by determining by determining the five indicators: new size, existing size, reimpl, and retest as below:-
Se=Nwsize+ExSize(0.4*Rdsgn+0.25*Rimpl+0.35*Rtst)
After computing Se the estimated effort is calculated as below:
$\text{Effort} = D^{0.4} * (Se/Cte)^{1.2}$
Where D= Relevant to the staffing aspects
Se= Effective size found in the previous equation.
Cte= effective technology.
Once effort is obtained, duration is solved using the following equation
$Td=D^{-0.2}*(Se/Cte)^{0.4}$.
Where Td = time duration.
This equation relates the effective size of the system and the technology being applied by the developer to the implementation of the system [14].
SEER-SEM has two main Disadvantages on effort estimation.

i) Input parameters of about 50 are related to the various factors of a project, thereby increasing the complex nature of SEER-SEM, especially for the management of uncertainty from outputs of these inputs.

ii) Specific details of SEER-SEM widening the difficulty of identifying the non-linear relationship between parameter inputs and corresponding outputs . Overall, these two major disadvantages leads to a lower precision in effort estimation by SEER-SEM [13].

b) *Putnam Model*: Putnam model is another popular software cost estimation model and the model form is :

Technical constant C= size*B1/3*T4/3
Total Person Months B=1/T4 *(size/C) 3
T= Required Development Time in years

Size is estimated in LOC
Where, C is a parameter dependent on the development environment and it is determined on the basis of historical data of the past projects.

Rating: C=2,000 (poor) , C=8000 (good) C=12,000 (excellent) .

Putnam model is very delicate towards the development time , the person months needed can be greatly increased by reducing the development time[2][4].
One important problem with the PUTNAM model is that it is based on knowing, or being able to estimate accurately the size (in lines of code) or cost of the software to be developed. There is often great uncertainty in the software size. Resulting in the imprecision of cost estimation[4]
Advantages: The simplicity with which the putnam model is calibrated is the main advantage available to this model . Data such as size , duration and effort can be easily collected for the previous projects by most of the software companies , regardless of the maturity level . Process productivity which is exponential in general is converted to linear index productivity so that changes in their own organization can be tracked and can be applied in future [2].
Disadvantages: Putnam model is based on knowing , or being able to estimate accurately , the size (in terms of lines of code) of the software to be developed which is one of the significant problem .The uncertainty of the in the software size is very often in this model. Sometimes it may also mislead in the precision or accuracy of the software cost estimation.

Duration(time) and size are only the two variables on which the putnam model is based upon and duration(time) being the dominating factor of the model. Rest of the aspects of the software development life cycle (SDLC) are not considered. Nearer results were obtained by COCOMO II because it considers almost all aspects of the software development life cycle (SDLC) [2].

S.NO	Type of cost estimation model (method)	Advantages	Disadvantages
1.	Putnam model	Based mainly on two variables that is size and duration (time).	All other aspects of the SDLC (software development life cycle aren’t being considered in this method).
2.	SEER-SEM	Software size as the important feature, and the effect size parameter.	More input parameters increasing the complexity of the project , difficulty in discovering the nonlinear relationship between input and corresponding output parameters.

Table 1: Strengths and Weakness of Algorithmic Models

B. Non-Algorithmic methods

Machine learning methods like Genetic algorithm, artificial neural network are used in the prediction of the software cost

List of non algorithmic models explained in this models are :

1) *Neural Networks*: Neural network is a powerful class of model which also includes many diverse models and approaches. The processing of the information by a neural network is alike the biological nervous system, similar to brain processing information. For processing the information it has a novel like structure. A large number of neurons (i.e densely interconnected processing elements) work together as one for solving a problem. Specific applications like data classification or pattern recognition, is configured by neural network through a learning process . Neural network learn by examples (i.e through giving both input and the output). They do not follow algorithmic approach [7]. The neurons are interconnected from one another through an interconnection link. Weights are associated with each link which contains information of the input signal. The activation level is called the internal state of the neuron. Many number of activation functions such as sigmoidal, tanh , guassian , linear etc , sigmoidal is the most frequent used activation function in the neural network [6].

Thus, the models of ANN are specified by the three basic entities namely [6]

a) The synaptic interconnections of the model

b) The training or learning rules adopted for updating and adjusting the connection weights.

c) Their activation functions.

The basic structure of the neural network is show below in the fig below (fig 1).

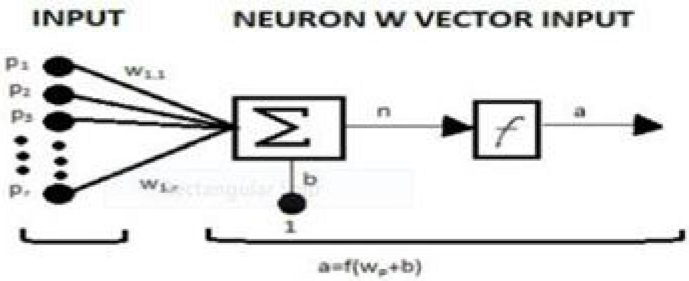

Fig 1: basic structure of a neural network [1].

2) *Fuzzy Logic:* Fuzzy set theory is the concept behind the fuzzy logic . It is a theory of classes with un-sharp boundaries, and it also considered to be an extension for classical set theory [1] . Fuzzy logic systems are mainly categorized into three types: pure fuzzy logic systems, Takagi and Sugeno’s fuzzy system, and fuzzy logic system with fuzzifier and defuzzifier. Fuzzifier and Defuzzifier which were first proposed by Mamdani are the most widely used fuzzy logic systems . These are successfully applied and tested on different types of consumer end products [1].

The membership μ a(x) of an element x of a classical set A, as subset of the universe X, is defined by [1]:

$$\mu_A(x) = \begin{cases} 1 & if \quad x \in A \\ 0 & if \quad x \notin A \end{cases}$$

Fig 2: The membership μ a(x) of an element x of a classical set A

The steps to be performed to estimate or predict the cost of the software are :

Step1: Fuzzification : conversion of the crisp input to the fuzzy output is done in this step.

Step2: Fuzzy Rule Based System: Fuzzy IFTHEN rules are used by the systems of fuzzy logic. After the fuzzification process the crisp input values are fuzzified into respective Linguistic values.

Step3: Defuzzification-process of converting the fuzzy output into crisp output [8].

3) *Genetic algorithms:* Genetic algorithms are heuristic search algorithms based on the Darwin theory of natural selection [10]. All possible solutions are searched by the genetic algorithm using population of the individuals which is consideredto be the

potential solutions for the problem under consideration. offsprings which are variations of their Parents are produced by the solutions which are the best fit to survive in the upcoming generations . Genetic algorithms have been successfully used in different types of difficult numerical optimization problems. They are successfully used for solving signal processing and path searching problems [10]. The evolutionary process starts by the computation of the fitness of each individual in the initial population. For stopping the evolutionary process we need to follow or perform the following steps.

a) Use the selection mechanism (i.e , tournament , roulette wheel , rank) for reproduction of the selected individuals.
b) Use the crossover and mutation operators to create an offspring. Based on the application the probability of the mutation and crossover are selected.
c) Compute the obtained new generation [10].

4) *Parkinson:* Using Parkinson principle the determination of cost is done by the resources available instead of based on an objective assessment. If the software has to be delivered in 12 months and 5 people are available, then the effort is estimated as 60 person-months. How ever this method gives good results sometimes bit, it often provides unrealistic estimates which is strongly not recommended and it also does not promote good software engineering practice [11][5].

5) *Estimation By Analogy* :Software cost estimation using Analogy is considered to be one of the most useful method for the cost estimation. Different types of cost estimation models have been developed based on this analogy method. The principle in which analogy is based on is choosing a identical or a similar previous project which are better indicators for the performance of the future project. Comparing the proposed project to the previously completed similar project where the project development information is known is called Estimation by anology. Actual data from the completed projects are used to to estimate the proposed project. This method can be used at system- level or at the component-level[4]. The following are the steps used by estimation of Anology:

a) Listing out the characteristics of proposed project.
b) Selection of most identical completed previous projects and the database contains its characteristics.
c) Estimating the proposed project from the most identical completed previous project by analogy[5].

The advantages of estimation by Analogy are:

a) The basis of estimation is the characteristic data of the actual project.
b) The estimator's knowledge of past experience can be used which is not easy to be estimated.
c) The differences between completed and proposed project can be estimated. [1]

Disadvantages in the Analogy based estimation technique are given below :

a) Determine the best way to describe the project using this method . The restriction for the choice of variables to information that is available at the point where prediction required. [1][4]
b) Possibilities are including the type of application domain, Referencing the number of distinct entities, the number of screens , the number of inputs.[1]
c) Even though after characterizing the project, one should determine the similarity and how much confidence that can be placed in the analogies. Very few analogies will lead to projects like maverick being used; Many will lead to the dilution of the impact of analogies. Standardization of values take place so each dimension contributes equal weight for finding analogies [4].
d) Finally, by knowing the effort values from the past projects we must derive an estimate for the proposed new project. means and weighted means include possibilities which are more nearer to the analogies[4].

The selection of the right set of the previous or past projects is considered to be the most important aspect fot the success of Analogy based cost estimation technique . Analogy based technique is considered to be superior when compared with the algorithmic models in certain circumstances . Analogy based estimation is also easy to learn and and estimate the cost [4].

6) *Price-to-win*: Price- to-win as the name depicts this model is used for winning the project which is the best price estimated . Instead of the software functionality the estimation is based on the customer’s budget . For example, if a reasonable estimation a customer can afford 60 person-month’s and the project estimation costs 100 person-months ,in order to win the project it is usual that the estimator is asked to change the estimation to fit 60 person month’s effort . This is not a good way of practicing

since it is very likely seems to cause a very bad delay of delivery or force the development team to work overtime[5].

S.NO	Type of cost estimation model (method)	Advantages	Disadvantages
1	Analogy	The estimation or the work is based on the actual characteristic of data and actual experiences , the differences between the proposed and the completed (past) projects can be identified .	There is a need of lot of information about the previous or the past products , in certain cases there may be no existence of the similar project in the database .
2	Fuzzy Logic	Provides an accurate estimate with the help of the fuzzy rules . No training is required and flexibility.	The complexity of the project increases as the complex of the fuzzy logic increase .
3.	Neural Networks	Consistent with the unlike databases , Power of reasoning .	The performance depends on large training data and there is no guidance for the designing .
4.	Parkinson law	Determination of the cost based on the available resources . No overspend.	Often providing unrealistic estimates , and not a good software practice , system usually unfinished
5.	Price-to-win	Winning the project or the contract chances are high .	Poor practices are being reinforced

TABLE 2: Strengths And Weakness Of Non-Algorithmic Models.

IV. CONCLUSION

Many projects have been failed in last few decades, one of the major reason behind a failure of a project is the inaccuracy of the cost estimated in the development of the project .The main objective of the paper is to provide detailed literature survey about the various existing cost estimation models. This paper includes many software cost estimation models including both the algorithmic and non-algorithmic models like putnam's model , seer-sem model , analogy , neural network , fuzzy model , parkinson , price to win . No method or model is perfect i.e in all the situations there exists no model for cost estimation with more precise outcome . Before choosing the estimation method it is very essential to know about the principle of each and every model ,before choosing the best .This paper mainly focuses on the strengths and the weakness of the various software estimation models for choosing the estimation model depending on the complexity and nature of the software project .Improvement in the performance of the existing methods and introducing the new methods for cost estimation based on current software project scenario and requirements can be the future works in this area.

REFERENCES

[1] Poonam rijwani , sonal jain and dharmesh satani,"Software Effort Estimation: A Comparison Based Perspective " , International Journal of Application or Innovation in engineering and management (IJAIEM) , Vol 3 , Issue 12 , December 2014 .

[2] Shubhangi Mahesh Potdar , Manimala puri , Mahesh P. Podtar , " Literature Survey on Algorithmic methods for Software Development Cost Estimation " IJCTA , Vol 5(1) , 183-188 , 2014 .

[3] S. Malathi and B. S. Lijin , " An Efficient Method for the Estimation of Effort in Software Cost " , International Journal of Advance Research in Computer Science and Management Studies , Vol 2 , Issue 2, Feb 2014

[4] Narendra Sharma , Aman Bajpai and Ratnesh Litoriya ," A comparison of software cost estimation methods: A Survey ",The International Journal of Computer Science & Applications (TIJCSA) , Vol 1 , No. 3, May 2012 .

[5] Sweta Kumari and Shashank Pushkar , "Performance Analysis of the Software Cost Estimation Methods: A Review ", International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) , Volume 3, Issue 7, July 2013.

[6] Anupama Kaushik, A.K. Soni & Rachna Soni , "A Simple Neural Network Approach to Software Cost Estimation" Global Journal of Computer Science and Technology Neural & Artificial Intelligence, Volume 13 Issue 1 ,Year 2013.

[7] Divya Kashyap , Ashish Tripathi and Prof. A. K. Misra , "Software Development Effort and Cost Estimation: Neuro-Fuzzy Model ",IOSR Journal of Computer Engineering (IOSRJCE), Vol 2 , Issue 4 , 2012 .

[8] Vidisha Agrawal and Vishal Shrivastava "Neural Network Approaches for Software Development Time Estimation: A Review" .

[9] Ghose,Mrinal Kanti,Roheet Bhatnagar and Vandana Bhattacharjee. "Comparing some neural network models for software development effort prediction." ,Emerging Trends and Applications in Computer Science (NCETACS), 2011 2nd National Conference on. IEEE, 2011.

[10] Brajesh Kumar Singh and A. K. Misra , "Software Effort Estimation by Genetic Algorithm Tuned Parameters of Modified Constructive Cost Model for NASA Software Projects ", International Journal of Computer Applications , Volume 59– No.9, December 2012.

[11] Hareton Leung and Zhang Fan, "SoftwareCostEstimation",https://www.st.cs.unisaarland.de/edu/empirical-se/2006/PDFs/leung.pdf

[12] Alaa F. Sheta , Aladdin Ayesh , David Rine "Evaluating software cost estimation models using particle swarm optimisation and fuzzy logic for NASA projects: a comparative study " , International Journal Bio-Inspired Computation, Vol. 2, No. 6, 2010 .

[13] Omprakash Tailor,Jyoti Saini and Poonam Rijwani, "COMPARATIVE ANALYSIS OF SOFTWARE COST AND EFFORT ESTIMATION METHODS: A REVIEW", International Journal of Computer Science and Mobile Computing , IJCSMC, Vol. 3, Issue. 4, April 2014, pg.1364 – 1374 .

[14] Lee Fischman, Karen McRitchie, and Daniel D. Galorath ,"Inside SEER-SEM",CROSSTALK The Journal of Defense Software Engineering ,april 2005

[15] Lalit V. Patil , Sagar K Badjate and S.D.Joshi,"Develop Efficient Technique of Cost Estimation Model for Software Applications", International Journal of Computer Applications (0975 – 8887) Volume 87 – No.16, February 2014.