Software Cost Estimation Methods: A survey

Gopal S. Chandak, Uddhav S. Holkar, Avinash B. Nalawade, Ulhas N. Chalawade

B.E. Student (2012-13),
Department of Computer Engineering,
Vidya Pratishthan's College Of Eng.,
Baramati-413133, Pune, Maharashtra,
INDIA.

Prof. Dinesh B. Hanchate
Department of Computer Engineering,
Vidya Pratishthan's College Of Eng.,
Baramati-413133, Pune, Maharashtra,
INDIA.

***Abstract*— Cost estimation for the software is one of the prime portions of any project. Correct estimation of cost for the software will help in completing project in stipulated interval of budget and time. Here we will be comparing of various methods of estimating cost of any software and estimation of some model, which could be utilized for the cost estimation by software of any project. This method gives an introduction sight of each of obtainable techniques and the cost estimation work with regards to data mining. The prime objective here is to have a comparable study of each of tools and techniques available.**

***Keywords*— Software Cost estimation, Software Cost Estimation Methods.**

I. Introduction

Here, we get an introduction view of upcoming professionals for giving details about the software models and methods for estimation of its cost. There are various models and methods that are been used for cost estimation of software, but it is too much difficult for deciding that which method is appropriate for cost estimation. In order to have a solution for such type of problem it's important to have knowledge regarding the cost estimation models and methods of software. This paper gives each of the techniques of cost estimation which could materialize its use in various environments [1]. In the research of years of recent past, plenty of cost estimation software method are available that includes bottom-up method and top-down estimating by analogy, expertise judgmental methods, algorithmic methods. There is no one method which we can says the necessity of worse or better than other. In actuality the weaknesses and strengths of those methods are mostly corresponding to one another. Whenever you want estimation for the projects, then it would be necessary to know and understand the weaknesses and strengths of the cost estimation method which we have to refer [2]. Making the correctness of the estimation of cost and the supplementary field might balance with the fields in software engineering. JPL for NASA has developed one of the cost estimation model named as 2CEE. 2CEE is the association of software engineering and data mining fields. Such estimation type could standardize the algorithms for machine learning with models for estimation of cost. The objective here is to do a research of all such methods and models and know the correct estimate of it.

II. SOFTWARE COST ESTIMATION MODELS AND METHODS

A. *EXPERT JUDGMENT METHOD*

It is nothing but an ability to do the prediction and then reside out or to move out from such problem in a do-main been given. The weaknesses and strengths are contradictory to the algorithmic method of weaknesses and strengths. To give the adequate wide bandwidth of communication for expertise to back-and-forth the amount of details that are required to standardize the estimation with respect to other expertise, a new technique known as wideband Delphi is been launched above Delphi technique which was considered as a standard [1]. The steps for estimating the use of wide band Delphi technique:

1. Each expert is presented with specification and estimation form by coordinator.
2. These forms are filled anonymously by Experts.
3. A group meeting is called by coordinator where the expertise discusses issues of estimation with the one another and the coordinator.
4. A summary of the estimation is distributed and pre-pared on an iteration form by Coordinator.
5. Expertise fill-out this forms anonymously again, and the 4th and 6th step are repeated plenty of times.

After passing four stages the eight experts which are contributed and the final convergence determined in Delphi technique (Mahmud S et al., 2008) is used which is shown example in Figure 1. The disadvantage includes It is difficult to document the factors utilized by the experts or experts-group.

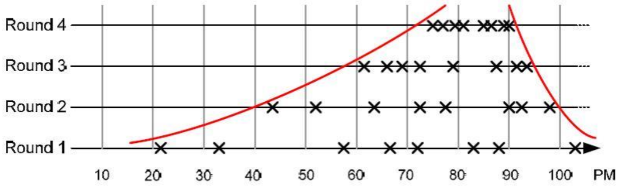

Fig. 1. An example using Delphi technique

This method could not be quantified. The expert judgment method mostly compliments the other cost estimating method such as algorithmic method [3]. Expert may be some optimistic, pessimistic, and biased although they would have been decreased by the group consensus.

B. *ESTIMATING BY ANALOGY*

Based on analogy methods for estimation process, this method is most useful as more models of cost estimation that are been developed. Software Estimation Based Analogy is formed on principles of actual efforts and values. Analogy methods for estimation could make its utilization at the component-level or at system-level [1]. In few aspects, it is proper forms of expertise judgments as the expert often do searching for matching situation and informing the opinions. The step for making use of estimation by analogy is as followed:

1. The proposed project is been characterized.
2. Making the selection of the most similar completed project whose characteristic had been store in an historical database.

To derive the estimation for a propose project by analogy from the most similar completed projects. In this method, function of similarity like Manhattan similarity (MS) Shepperd and Euclidean similarity (ES) and defined which compares features of two projects.

$$\mathrm{Sim}(p,\ p') = \frac{1}{\sum_{i=1}^{n} \sqrt{Dist\,(f_{i;}\ ,f_i')\times W_i}} \qquad (1)$$

Where p and p' are the project W_i is the weight varies in range of 0 to 1 for n features. f_i, and displays ith features for every projects. is 0.0001 and utilized for getting the non-zero result. ES and MS formulas are comparatively more same manner but it helps in computing the difference among them.

Advantages of estimating by analogy methods are

1. Estimating depends on the characteristic data of actual project.

2. The knowledge and past experience of estimator's could be utilized which is not simple to be quantify.

3. The difference among proposed project and completed one could identifies and estimated impacts.

III. TOP-DOWN AND BOTTOM-UP METHODS

is also known as Macro Model. An overall cost estimation for the project is on an estimation from the universal property of the software project by using top-down estimating method, and the project is divided in the various low level component or mechanisms. Putnam model is an approach for such a method. At early stage of the development of software, it is of too much of utilization since there is no adequate available information [1]. Remuneration includes activities of system-level like configuration management, documentation, integration, project control, etc. From that this might be ignoring in other methods of estimation. Top-down method is faster generally. It is also simple for implementation and it require the minimum de-tail of project. Disadvantage is that it could be having less accuracy and tend to neglect the lower level component and the possibility of technical problem.

Advantages are:

1. It requires minimum details of project. It is faster generally and simple for implementation.
2. It focused on activities of system level such as con-figuration management, integration, documentation, some of which might get ignores in other methods of estimation and it wouldn't miss the system level function cost.

Dis-advantages are:

1. It doesn't provide the details for justifying the estimate or decisions.
2. It doesn't identifies the difficulty of low level problem which could escalates cost and sometimes tend to over-look low level component.

IV. ALGORITHMIC METHOD

This method model gives the importance of the algorithm based model in the files of estimation of software cost. Various cost estimation software method makes use of algorithmic method which is classified into different models. For estimating the software cost each model uses the equation:

$$\mathrm{Efforts} = \mathrm{f}(\mathrm{x1},\ \mathrm{x2}, \ldots,\ \mathrm{xn}) \qquad (2)$$

Where the vector of the cost factor is $(\mathrm{x1},\ \mathrm{x2}, \ldots,\ \mathrm{xn})$. The equation is dependent on the historical and research data and makes utilization of input as design methodology, Source Lines of Code (SLOC), risk assessments, skill levels, number of functions to perform and other cost drivers such as language etc. The algorithmic method is design so as to give many mathematics equations to perform the estimation of software. These equations of mathematics are dependent on historical and research data and make utilization of input such as design methodology, Source Lines of Code (SLOC), risk assessments, skill levels, number of functions to perform and other cost drivers such as language etc. Various models has been develop and the algorithmic methods has been studied Largely, such as Putnam model, COCOMO models, and models based on function points [1] [4].

Advantage:

1. It is easy to modify input data and also refine as well as customize formulas.
2. It is able to generate repeatable estimations.
3. It is objectively calibrated to previous experience.
4. It is efficient and able to support a family of estimations or a sensitivity analysis.

Disadvantage:

1. Poor sizing inputs and inaccurate cost driver rating will result in inaccurate estimation.
2. Some experience and factors cannot be easily quantified
3. It is unable to deal with exceptional conditions like exceptional teamwork, exceptional personnel in any software cost estimating exercises, and an exceptional match between skill-levels and tasks.

V. COCOMO MODELS

Constructive Cost Model (COCOMO) is the algorithmic software cost model. In very simple form basic COCOMO model is given [1] [5]:

$$\mathrm{E} = \mathrm{K1} \times KLOC^{K2} \qquad (3)$$

$$\mathrm{D} = \mathrm{K3} \times (\text{Efforts Applied})^{K4} \qquad (4)$$

$$\mathrm{P} = \frac{Effort\ Applied}{Development\ Time} \qquad (5)$$

Where D is development time expressed in terms of months, E is effort applied expressed in man-months, P is count of people required, The number of estimated delivered lines (in thousands) of code for the project is by KLOC. The co-efficient K1, K2, K3, and K4 dependents on application and environment development. The co-efficient of K1, K2, K3 and K4 are mentioned in following table as:

TABLE I
Comparison with other Heuristics

Software Project	K1	K2	K3	K4
Semi-detached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32
Organic	2.4	1.05	2.5	0.38

By making an estimation from the simple model of CO-COMO could be done more precisely by considering into an account the other factor that concerns the needed characteristic of software that needs to be developed, the experience and qualification of team of development as well as the environment for development of software. There are few factors which are given below:

1. Required efficiency (memory and execution time)
2. Database Size
3. Required reliability
4. Experience of team in the application area
5. Analyst and programmer capability
6. Use of tools and software engineering practices
7. Experience of team with the programming language and computer.
8. Software Complexity.

These are some of the factor that affects the person months needed by the order of enormity or more. Constructive Cost Model assumed that the software and system requirements should be define, and such requirement is irreversible one [4]. Constructive Cost Model is a model of regression. It relies on the analysis of selected 63 projects. KDSI is the primary input (thousands of delivered source instructions) [6] [5].

VI. PUTNAM MODEL

Equation of Putnam's Model is as follows:

$$\mathrm{S} = \mathrm{E} \times Efforts^{1/3} \times t_d^{4/3} \qquad (7)$$

Here, t_d is the time of delivery, E is the indication for environment and ability for demonstrating the environment; S and Effort are express by person year and LOC respectively. And Efforts are calculated by the formula:

$$\text{Effort} = D_0 \times t_d^3 \qquad (8)$$

Here, D_0 is build-up factor for manpower, which gets verified from 8 (new software) to 27(rebuilt software). The model of Putnam is been delicate to time of development by declining a time for development could largely increases development of the person months that are required. The main difficulty with model of PUTNAM is it relies on knowledge or to get estimation of cost accurately for the size (in LOC) of development of software.. Due to high unreliability in the size of software it might result in an erroneous estimation of cost.

VII. AGILE COCOMO MODEL

A COCOMO incorporates the full parametric model for COCOMO and is utilized for the estimation based on analogy in order to get the correct result for the latest projects. The most significant way for an software cost and its effort estimation is by analogy based estimation. In doing comparison of the likeness among old and new project gives a significant path for estimation, outcome can however be not in that accurate from gathering the difference among two project is moreover if the ground of dis-similarity is of more importance. Building estimation by approach of analogy that accounts the difference among the project, Agile COCOMO-II model has been invented by USC-CSE, a tool for estimation of cost which rely on model of COCOMO-II. The estimation used here is analogy-based to produce the correct result that been quite easier for its use and easy to learn. It can give the facility to estimate the project in various ways; its given in Figure 5. We could make estimation for project in terms of person month; object points, dollar, and function points, etc. Here we have made an discussion for the simulation for program, the results of our research, structure of program, and gives inner view of this tools in to the upcoming way [2].

VIII. FUNCTION POINT ANALYSIS BASED METHODS

Another method for the size quantifying and software system complexity with regards to function which is delivered to user by the system is done by a method called as function point analysis. A measurement unit that expresses the functionality and amount of information for a business system that is provided to a user is nothing but a Function point. The cost (in $ or hour) for the single unit is done on the calculation from the projects done previously. The method for measurement of function point was published in 1979 and created at IBM by Allan Albrecht.

The estimate for the cost is done by the determination of indicators following:

1. Logic Files
2. Input from User
3. Output from User
4. Interfaces
5. Enquiries

For each indicator Degree of Complexity is defined as 1, 2 and 3 stands for simple, medium and complex respectively. For each indicator weight is defined between 3 and 15.

The equation for that is given below:

$$\text{UFC} = \sum_{i=1}^{5} \sum_{j=1}^{3} N_{ij} \times W_{ij} \quad (9)$$

Where, UFC stands for unadjusted function point count, W_{ij} is the weight of indicator i with complexity j and $N_{ij}$is the number of indicator i with complexity j. The advantages:

1. Function points are independent of the language, methodologies used for implementation or tools.
2. The estimation of function points could be made from specification of design or requirement specification, hence possibly making estimation for cost of development in an early phase of development.

IX. THE SELECTION OF ESTIMATION METHODS

Here we have learned various models for estimation. Researchers continue the working on process of software cost estimation for improving the accuracy. From the comparison of methods, we came to know that there is not an single method that is good or bad from one another, in fact, their weaknesses, strengths, are often admiring to one another. Accordance with the knowledge and understanding, it is commended that a union of expertise or analogy discernment models and methods of estimation is handy to get authentic, and to the correct estimation of cost for development of software. If most part of project is same to the previous projects that places analogy method or judgment of expert will be handy. For such estimation type method of analogy generate the more correct results than other method as its quite reliable and fast under such circumstance; so its good to make utilization for model of algorithm for projects known lesser or larger. The model of estimation does not requires its input as SLOC as recommended by many researchers. After studying all the models, model for agile COCOMO generates the more correct

cost of projects than the other models available as that predicts the project cost in diverse way could be cost of the project in $, function points ,in terms of person month, object points etc. It is very difficult task for selecting the correct estimation. Even for a small fault for a task, the outcome will be of huge loss considering the financial status and increase in time for project completion. This is usual that we will be applying few methods of cost estimation for estimating development cost of software. It is quite important to for the continue re-estimating of cost and comparing target opposed to authentic disbursement at every important level, i.e., the milestone. This will help in keeping the status of project visible and helpful in identifying the required correction for schedule and budget as it occurs.

Advantages are as following:

1. Various biases and viewpoints could be considered and its reunite. High business priority to bring costs down, a competitive contract bid, or a window of small market with the outcome of strict deadline tend to be hopeful estimate.
2. Various method of estimation might utilize disparate data for processing. These outcomes in good cover of the base knowledge for a process of estimation which helps in identifying the components cost which can't be given out with or were ignored in any of method.

X. CONCLUSION

Various cost estimation method for software is available that includes analogical estimation, algorithmic method, method of bottom-up, expert judgment method and top down method. From the comparison of methods, we came to know that there is not a single method that is good or bad from one another, in fact, their weaknesses, strengths, are often admiring to one another. Which estimation method to be used for a particular project estimation that depends on project nature. As per the weaknesses or strengths of the method, you could make a choice regarding what method to make use of at the moment. We only need to insert value for various drivers as per cost with regards to data of previous project. Other COCOMO models can provide all facilities. It generates the new project cost in much correct manner than various models of cost estimation. The upcoming task is to learn latest cost estimation models and methods of software which could be helpful for easy understanding of the process of estimating software cost.

References

[1] A. AJ., \Measuring application development productivity.," Proceedings of the joint SHARE/GUIDE and IBM Application Development Symposium. SHARE, Inc. and GUIDE Intl. Corp.: Chicagho IL, pp. 83{92, 1979.

[2] B. Bohem, Software Engineering economics. New Jersey: Prentice Hall PTR, 1981.

[3] A. J. Albrecht and J. E. Gaffney, Software function, source lines

of code, and development effort prediction: A software science validation," IEEE Trans. Softw. Eng., vol. 9, pp. 639{648, Nov. 1983.

[4] L. C. Briand, V. R. Basili, and W. M. Thomas, \A pattern recognition approach for software engineering data analysis," IEEE Trans. Software Eng., vol. 18, no. 11, pp. 931{942, 1992.

[5] F. J. O. R. S. C. Breiman, L., Classi_cation and Regression Trees. Wadsworth & Books/Cole Advanced Books & Software, 1984.

[6] C. T. Bohr Albrecht, A.J.nstedt G., Robustness in Regression Analysis. In: Costner, H. Chapter 5. Sociological Methodology, Jossey-Bass, 1971.