REVIEW ON SOFTWARECOSTESTIMATION

Gajanan Chakote

Head
Computer Science and Engineering
Matsyodari Shikshan Sanstha, Jalna India.

Abstract: Software cost estimation is the process to predict the complete cost required to develop software. Many estimation techniques have been proposed since last 3 decades. The software cost estimation is mostly depends upon the estimation of software efforts and time required to develop the software. Accuracy of software cost estimation is completely depends upon how accurately the size and effort should calculated. Software cost estimation techniques.

Mostly classified in algorithmic, non- algorithmic and machine learning. Each has its own specific strengthsand weaknesses. As election factoring cost estimation technique sis the accuracy of its estimates. In most of the cases the pre-experience with estimation model following with similar kind of software project is consider as a prime factor for to estimates the cost of software systems. The accuracy of the semodelsis not satisfactory.

The objective of this paper is providesa general overview of various relevant software cost estimation techniques with their aspects. Paper includes various cost estimation techniques discussed with their performance and their uses.

Index Terms **-Project estimation, Effort estimation, Cost model, Cost accuracy, Prime factors.**

I. INTRODUCTION

If we see the current scenario of increased trends in hardware and software cost comparison then software cost are increasing tremendously. We found in recentyears,softwarehasbecome one of themost essential andexpensiveelementofcomputersystem projects. The cost estimation process starts from planning of software project. Estimating the software project is initiated by measuring the effort required for a project actually it is measure in human workforce required to complete the project successfully. It is measure in human workforce required per day , per month or per year basis to completely develop the software project. Themajor part ofthecostofsoftwaredevelopmentisduetothehumaneffort,hence most of the cost estimation methods focus on this aspect and give estimates in terms of person-months.

Estimates the accurate software cost is one of the mostcritical part for bothdevelopersandcustomers. Underestimates increases the extra risks and burden on developers and management whereas overestimates causes software failure or unsatisfaction of customers.

An efficient and accurate cost estimation methodology is very essential for software developer team which will assist the management to estimate the cost.

In this paper we discussed the various current available techniques used to estimate the cost. The paper begins with the introduction followed by the literature review in second. The third section of the paper includes the methodology followed by the conclusion.

II. Cost Estimation Techniques

The various available software cost estimation techniques are mostly classified into three main categories on abstract way. 1. Non algorithmic 2. Algorithmic techniques and 3. Machine learning techniques.

Non algorithmic cost estimation techniques mostly follow the historical available data for similar kind of projects. Non algorithmic techniques commonly use the expert judgement methods and estimation by analogy categories. Whereas machine learning techniques use the training rules and repeat the cycles to estimates the software costs. The techniques used in machine learning type are fuzzy logic and neural networks. Whereas estimating software cost by the algorithmic techniques includes assumptions and calculation to estimate the size of the project first. Estimating the size of the project is followed by estimating the efforts required by software to get it done successfully. The required efforts are measures in terms of human workforce required per month. Finally the cost of software project is calculated by calculating the required cost needed to satisfying

required human workforce.

1. Non algorithmic methods

Expert judgment

Expert judgment method is most commonly used in non algorithmic cost estimation techniques;in expert judgment techniques the similar kind of historical data of is analyzed to predict the cost of proposed software projects.

a) Wideband Delphi technique

The wideband Delphi techniques is advance method of Delphi techniques where most of the part is focus on providing the broad communication bandwidth to experts for to exchange the volume of information necessary to expert to calibrate cost estimations.

Estimation process steps

1. Specification and Estimation forms provide to experts by coordinators
2. Coordinators call a group of meetings to discuss issues in cost estimation with experts and each others.
3. Coordinator fills out the forms anonymously.
4. Coordinators prepare a summary of estimate and distributes among expert with iterative forms.
5. Variations among the expert view has identified coordinator calls meeting to discuss the widely varies estimates to resolve it.
6. Experts fill out forms, again anonymously, and steps 4 and 6 are iterated for as many rounds as appropriate.

The advantages of this method are:

1. The difference in requirements of the proposed project and past project experience can be factor by experts,
2. The experts can factor in project impacts caused by new technologies, architectures, applications and languages involved in the future project and can also factor in exceptional personnel characteristics and interactions, etc.

Method includes disadvantages as:

1. Method cannot be quantified not be scalable.
2. Hard to document the factors used by the experts or experts-group.
3. Expert may be some biased, optimistic, and pessimistic, even though they have been decreased by the group consensus.
4. This method always compliments the other cost estimating methods such as algorithmic method.

b) Estimating by Analogy

Estimating by analogy means comparing the proposed project to past and previously completed similar kind of project where the project development information is known. Actual data from the completed projects are extrapolated to estimate the proposed project. This method can be used either at system-level or at the component-level.

It is relatively straightforward method . in some respects, it is a systematic form of expert judgment since experts often search for analogous situations so as to inform their opinion.

Estimation process steps are:

1. Characterize the proposed project mostly based on requirements and its specifications.
2. Select the most similar completed projects whose characteristic matches with the proposed system which have been stored in the historical data base.
3. Derive the estimate for the proposed project from the most similar completed projects by analogy.

The advantages of this method are:

1. Mostly the estimation is based on actual project characteristic data.
2. The estimator's past experience and knowledge can be used which is not easy to be quantified.
3. The differences between the completed and the proposed project can be identified and impacts estimated.

Disadvantages are

1. The choice of variables must be restricted to information that is available at the point that the prediction required.
2. Even once we have characterized the project, we have to determine the similarity and how much confidence can we place in the analogies. Too few analogies might lead to maverick projects being used too many, might lead to the dilution of the effect of the closest analogies.
3. Finally, we have to derive an estimate for the new project by using known effort values from the analogous projects. Possibilities include means and weighted means which will give more influence to the closer analogies.

Top-Down and Bottom-Up Methods

a) Top-Down Estimating Method

Top-down estimating method is also called Macro Model. Using top-down estimating method, an overall cost estimation for the project is derived from the global properties of the software project, and then the project is partitioned into various low-level components. The leading method using this approach is Putnam model. This method is more applicable to early cost estimation when only global properties are known. In the early phase of the software development, It is very useful because there are no detailed information available.

The advantages of this method are:

1. It focuses on system-level activities such as integration, documentation, configuration management, etc., many of required functions which may be ignored in other estimating methods and it will not miss the cost of system-level functions.
2. It requires minimal project detail, and it is usually faster, easier to implement.

The disadvantages are:

1. It often does not identify difficult low-level problems that are likely to escalate costs and sometime tends to overlook low-level components.
2. It provides no detailed basis for justifying decisions or estimates.

Because it provides a global view of the software project, it usually embodies some effective features such as cost-time trade off capability that exists in Putnam model.

b) Bottom-up Estimating Method

Using bottom-up estimating method, the cost of each software components is estimated first and then combines the results to arrive at an estimated cost of overall project. It aims at constructing the estimate of a system from the knowledge accumulated about the small software components and their interactions. The leading method using this approach is COCOMO's detailed model.

The advantages are:

1. It permits the software group to handle an estimate in an almost traditional fashion and to handle estimate components for which the group has a feel.
2. It is more stable because the estimation errors in the various components have a chance to balance out.

The disadvantages are:

1. It may overlook many of the system-level costs (integration, configuration management, quality assurance, etc.) associated with software development.
2. It may be inaccurate because the necessary information may not available in the early phase.
3. It tends to be more time-consuming.
4. It may not be feasible when either time or personnel are limited.

Price to win

This method is mainly based upon the actual budget of customer's instead of the software functionality. The software cost is estimated by keeping objective in a mind to make is successful and complete software project in available price. If a reasonable

estimation for a project costs 100 person-months but the customer can only afford 60 person-months, it is common that the estimator is asked to modify the estimation to fit 60 person-months efforts in order to win the project.

The Advantages

1. Estimation cost is according to the customer's choice.
2. Less choices of failure.

The Disadvantages are:

1. It is very likely to cause a delay in delivery or force the developer's team to work overtime.
2. Developers may suffer from loss.

2. Algorithmic Method

The algorithmic method is designed to provide some mathematical equations to estimates the software cost. These mathematical equations are based on research and historical data and use inputs such as no. Of Source Lines of Code (SLOC), number of functions to perform, and other cost drivers such as language, design methodology, skill-levels, risk assessments, etc.

The algorithmic methods have been largely studied and there are a lot of models have been developed, such as COCOMO models, Putnam model, and function points based models.

Advantages of algorithmic model:

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

General disadvantages:

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

COCOMO Model

One very widely used algorithmic software cost model is the Constructive Cost Model (COCOMO). The basic COCOMO model has a very simple form:

MAN-MONTHS = K1* (Thousands of Delivered Source Instructions) ^{K2}

Where K1 and K2 are two parameters dependent on the application and development environment.

Estimates from the basic COCOMO model can be made more accurate by taking into account other factors concerning the required characteristics of the software to be developed, the qualification and experience of the development team, and the software development environment. Some of these factors are:

Complexity of the software

1. Required reliability
2. Size of data base

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

Many of these factors affect the person months required by an order of magnitude or more. COCOMO assumes that the system and software requirements have already been defined, and that these requirements are stable. This is often not the case.

The advantages are:

1 Simple to estimate the cost of project.\

COCOMO model is a regression model. It is based on the analysis of projects. The primary input is KDSI.

The disadvantages are:

1. In early phase of system life-cycle, the size is estimated with great uncertainty value. So, the accurate cost estimate cannot be arrived at last.
2. The cost estimation equation is derived from the analysis of projects. It usually has some problems outside of its particular environment. For this reason, the recalibration is necessary.

The first version of COCOMO model was originally developed in 1981. Now, it has been experiencing increasing difficulties in estimating the cost of software developed to new life cycle processes and capabilities including rapid development process model, reuse-driven approaches, object-oriented approaches and software process maturity initiative.

For these reasons, the newest version, COCOMO 2.0, was developed. The major new modeling capabilities of COCOMO 2.0 are a tailor able family of software size models, involving object points, function points and source lines of code; nonlinear models for software reuse and reengineering; an exponent-driver approach for modeling relative software diseconomies of scale; and several additions, deletions, and updates to previous COCOMO effort-multiplier cost drivers. This new model is also serving as a framework for an extensive current data collection and analysis effort to further refine and calibrate the model's estimation capabilities.

Putnam model

Another popular software cost model is the Putnam model. The form of this model is:

Technical constant $C = size * B^{1/3} * T^{4/3}$

Total Person Months $B = 1/T^4 * (size/C)^3$

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).

The Putnam model is very sensitive to the development time: decreasing the development time can greatly increase the person-months needed for development.

One significant problem with the PUTNAM model is that it is based on knowing, or being able to estimate accurately, the size (in lines of code) of the software to be developed. There is often great uncertainty in the software size. It may result in the inaccuracy of cost estimation.

The advantages are:

1. It considered the variable as a time and size, so accuracy increases.

The disadvantages are:

1. It doesn't consider the other aspects of the software development life cycle.

Function Point Analysis Based Method

From above two algorithmic models, we found they require the estimators to estimate the number of SLOC in order to get man-months and duration estimates. The Function Point Analysis is another method of quantifying the size and complexity of a software system in terms of the functions that the systems deliver to the user. A number of proprietary models for cost estimation have adopted a function point type of approach, such as ESTIMACS and SPQR/20.

The function point measurement method was developed by Allan Albrecht at IBM and published in 1979. He believes function points offer several significant advantages over SLOC counts of size measurement.

There are two steps in counting function points:

1. Counting the user functions. The raw function counts are arrived at by considering a linear combination of five basic software components: external inputs, external outputs, external inquiries, logic internal files, and external interfaces, each at one of three complexity levels: simple, average or complex.. .The sum of these numbers, weighted according to the complexity level, is the number of function counts (FC).
2. Adjusting for environmental processing complexity. The final function points is arrived at by multiplying FC by an adjustment factor that is determined by considering 14 aspects of processing complexity. This adjustment factor allows the FC to be modified by at most 35% or -35%.

The collection of function point data has two primary motivations. One is the desire by managers to monitor levels of productivity. Another use of it is in the estimation of software development cost.

There are some cost estimation methods which are based on a function point type of measurement, such as ESTIMACS and SPQR/20. SPQR/20 is based on a modified function point method. Whereas traditional function point analysis is based on evaluating 14 factors, SPQR/20 separates complexity into three categories: complexity of algorithms, complexity of code, and complexity of data structures. ESTIMACS is a propriety system designed to give development cost estimate at the conception stage of a project and it contains a module which estimates function point as a primary input for estimating cost.

The advantages of function point analysis based model are:

1. Function points can be estimated from requirements specifications or design specifications, thus making it possible to estimate development cost in the early phases of development.
2. Function points are independent of the language, tools, or methodologies used for implementation.
3. Non-technical users have a better understanding of what function points are measuring since function points are based on the system user's external view of the system.

The disadvantages are:

1 It is time consuming because it require manual work.

. 2 It requires pre-experience so, it is difficult for new developers

Machine Learning

The machine learning method for to estimate the software cost is another different way to predict the estimated cost required for software projects. In machine learning approach a training set ie. Hypothesis is provided and it is used to train the machine and response accordingly to learn the things as a artificial intelligence. To estimate the cost machine learning approach utilizes the training rules for estimations and repeat the run cycles. Hence it should be one of the appropriate approaches which increase the accuracy of the estimation's.

1. **Neural network**

The neural network approach is followed in wide area of research to recognize characteristic and improve the performance of the various system. The neural network approach is model as per the human neurons response to brain to communicate and responses to the reaction happen internally as well as externally with human.

The neural network approach is also used to estimate the cost of the software project where each neuron can represent the activity for a project and all these neurons gives output which is based upon the input they received for particular output. The overall output gives the entire cost estimation for a software projects. The neural network approach is very easy to learn as they modify the weights every time they used .this leads to better estimates as project leads to completion. The various methods are available in neuron network approach but in all these back propogation method is more accurate.

The advantages of this method are:

1. The estimation based on this method are consistent with unlike database , which provide better reasoning in estimation process.

The disadvantages are:

1 The large no. Of training data set is required.
2 No guidelines or instruction are provided for designing.

2. **Fuzzy Method**

All systems, which work based on the fuzzy logic try to simulate human behavior and reasoning. In many problems, which decision making is very difficult and conditions are vague, fuzzy systems are an efficient tool in such situations. This technique always supports the facts that may be ignored.

There are four steps in the fuzzy approach:
1.Fuzzification: to produce trapezoidal numbers for the linguistic terms.
2. To develop the complexity matrix by producing a new linguistic term.
3. To determine the productivity rate and the attempt for the new linguistic terms.
4. Defuzzification: to determine the effort required to complete a task and to compare the existing method.

The advantages of this method are:

1 No training set is required , also method is flexible to use.
2 It provides reliable estimates.

The disadvantages are:

1 This method is difficult to use.
2. Cost estimation of complex features is tedious.

IV.Conclusion:

Following the objectives that were set during the onset of this review, it was necessary to conduct a survey to get reliable and precise information for to estimate software. It is found in a survey that no single method to estimate the software cost is précised and accurate to proceed for different Kind of software projects.

Every software cost estimating techniques has its own strength and weakness which are useful as per the requirements requests generated by the customer. So it is essential to use the different available cost estimation tech. As per the projects requirements and necessities.

From the above review, we know no one method is necessarily better or worse than the other, in fact, their strengths and weaknesses are often complimentary to each other. According to the experience, it is recommended that a combination of models and analogy or expert judgment estimation methods is useful to get reliable, accurate cost estimation for software development

REFERENCES

[1] Alaa F. Sheta and Sultan Aljihdali " Software Effort Estimation Inspired by COCOMO and FP Models : A F Uzzy Logic Approach " IJACSA Vol.4 No. 11, 2013.
[2] Sonam Bhatia, Varinder Kaur Attri " Machine Learning Techniques In Software Effort estimation using COCOMO Dataset" IJRDO Volume-1 issue -6june, 2015, paper 13.
[3]Shivangi Shekhar, Umesh Kumar " Review of Various Software Cost Estimation Techniques" IJCA Volume 141 – No.11, May 2016.
[4]Boehm, " Software Engineering Economics ", Prentice Hall 1981.
[5] Gupta,Syona, Geeta Sikka and Harsh Verma,"Recent methods for software effort estimation by analogy",ACM SIGSOFT 'Software engineering Notes, 2011.
[6]Abedaillah Zaid,Mohd Hassan Selamat, Abdual Azim Abd Ghani, Rodziah Atan and Tieng Wei Koh" Issues in software Cost Estimatikon" IJCSNS ,Vol.8 No.11, November 2008.
[7]M. Jorgensen, M, Sheperd," A Systematic Reviews of Software Development Cost Estimation Studies"Software Engineering, IEEE Transactions on, Vol.33. pp 33-53. 2007.
[8]Vahid khatibi, Dayang N.A. Jawawi "Software Cost Estimation methods A review" JETCIS Volumw 2 No.1, 2010-11.
[9]Syed Zaffar Iqbal, Muhammad Idrres, Ahmed Bin Sana, Nawab Khan. Comparative Analysis of common Software cost Estimation Modelling Techniques. Mathematical Modelling and Applications. Vol.2. No.3. 2017.
[10] E. Miranda, "Improving subjective estimates using paired comparisons," *Software, IEEE*, no. February, pp. 87-91, 2001.
[11]Barry Boehm, Chris Abts and Sunita chulani " Sortware 'development cost estimation approaches – A survey ".

Annals of software Engineering 10(2000)177-205.
[12]Liming,"the comparison of the software cost estimating methods" http://www.compapp.dcu.ie/~renaat/ca421/lwu1.ht ml .
[13]Rekha Tripathi and Dr. P.k. Rai " Comparative Study of software Cost Estimation Techniquies"IJARCSSE Vol. 6,issue 1,Jan 2016.
[14]Shivangi Shekhar and Umesh Kumar" OKCA Vol.141, -No.11, May 2016.
[15] Attarzadeh, I.Siew Hock Ow, " Proposing a New Software Cost Estimation Model Based on Artificial neural Networks". IEEE International Conference on Computer Engineering and Technology (ICCET) Vol.-3, page(s) 487-49, 2010.
[16]Attarzadeh, I Siew Hock ow, " improving the accuracy of software cost estimation model based on a new fuzzy logic model", World Applied Science3 Journal 8(2), 117-184.2010.
[17]Shruti Jain, "Survey of Various Cost Estimation Techniques", International Journal of advanced Research in computer Engineering and technology(IJARCET), Volume , Issue -7, September ,2012.
[18]M. Jorgensena , "Review of studies on Expert Estimation of software Development Effort," Journal of System and Software , 70(1-2)37-60,2004.