Review of Various Software Cost Estimation Techniques

Shivangi Shekhar
Research Scholar-M.Tech-CSE
Govt. Women Engg.College,
Ajmer,Rajasthan

Umesh Kumar
Assistant Professor(CSE &IT)
Govt.Women Engg.College,Ajmer,Rajasthan

ABSTRACT
The industry of software should be efficient. Due to rapid change in technology, implementation of complex software systems at cheaper cost and the urge to maintain better quality software are some of the major challenges for the software companies. One of the toughest works is cost estimation, in the field of software engineering. It is the estimation of total cost required in developing software. Researchers have proposed various methods of cost estimation. This paper gives an insight into the various models and techniques used in estimating cost of the software. The benefits and drawbacks of the existing cost estimating techniques have been highlighted in this paper. There is as such not any single method which can be regarded as the best method so in this paper it is suggested that a combination of the methods should be used to get an accurate cost estimate.

General Terms
Software Engineering: Software Cost Estimation

Keywords
Software cost estimation, Estimation techniques, Cost models, COCOMO, Algorithmic method, Non-Algorithmic method.

1. INTRODUCTION
Cost estimation includes the process or methods that help us in predicting the actual and total cost that will be needed for our software and is considered as one of the complex and challenging activity for the software companies. Their goal is to develop software which is cheap and at the same time deliver good quality. Software cost estimation [1] is used basically by system analysts to get an approximation of the essential resources needed by a particular software project and their schedules. Important parameters in estimating cost are size, time, effort etc. Process of software estimation basically focuses on four steps. At first we estimate software size, then the needed effort after this we derive the schedule and at last calculate cost of the software. Various techniques are used in software cost estimation and we can broadly classify these techniques into two categories namely Algorithmic and Non-Algorithmic techniques. Algorithmic techniques rely on mathematical equations to estimate software cost. Constructive Cost Model (COCOMO) is a popular and widely used Algorithmic set of models. Algorithmic methods have many advantages but at the same time these methods are hard to learn and too much data are needed about the current project state in these methods. Conversely Non-Algorithmic techniques are easy to learn but we need to have complete information about one of the very similar previous projects as compared to our current software project, as estimation in these are made on the basis of historical data [2] .So basically there is as such not any single method which can be regarded as the best one. So a combination of the methods is usually suggested to arrive at a better cost estimate.

This paper begins with the introduction. Section 2 covers the various existing and emerging cost estimation techniques and their respective advantages and disadvantages. Section 3 deals with the selection of the estimation techniques. Section 4 is the conclusion and the references have been outlined in section 5.

2. COST ESTIMATION TECHNIQUES
Various methods are available to estimate the cost of software [3]. These techniques are classified mainly into two types: Algorithmic and Non-Algorithmic techniques [4]. In this section we enlist these techniques along with their advantages and disadvantages so as to get an idea which one is more suitable or can be regarded as the best technique.

2.1 Non-algorithmic techniques
Non-Algorithmic techniques base their estimation process on analogy and deduction. We need to have knowledge of a previously complete project similar to our current software project. Estimation is done on the basis of analysis of previous software projects or data sets. Some of the techniques based on Non-Algorithmic methods are as detailed below:

2.1.1 Estimation based on analogy
The basic idea behind estimation by analogy is that [5] whenever we get a new software project for cost estimation, it is just compared to historical similar projects to arrive at the nearest similar software project through which we can estimate our current project cost. The values and data from previously complete projects are deduced to calculate cost of our current project. We can use this technique both at system or component level [6]. The details of the exact steps followed for estimation based on Analogy are given below:

i. Determining the attributes of our current project.

ii. Finding a historical similar project as compared to our current project whose attributes are already stored in the database.

iii. Calculating the cost of the current project from the historical similar project.

Following enlists the advantages and disadvantages of this technique:

Advantages:

- Depends on the values and data of previous projects.
- Estimators experience can be used which helps in arriving at a better cost estimate.
- We get to know the minute distinction between the previous completed projects and our current projects and this in a way also helps in knowing their impacts.

Disadvantages:

- Using this method requires estimators to find out the attributes through which a project can be described best also we need to provide weightage to these to get a better analogy.
- We cannot use this technique for every project.

2.1.2 Expert judgement method

Estimation based on Expert judgement [7] captures the knowledge of experts and the estimation of cost is dependent on those projects which involved the inclusion of the expert. Usually there are some scenarios when we have limitations to gather and find data. Expert Judgement method is good to be used in these situations. It is the widely used estimation strategy for software projects. Wideband Delphi technique [8] is one of the cost estimation technique based on expert judgement. Here the participants are involved in two assessment rounds. Work breakdown structure is another example of the expert judgement method.

Advantages:

- The impacts caused due to new technologies, architecture and languages can be predicted by the experts.

Disadvantages:

- It is difficult and tedious to document the factors used by experts.
- We cannot fully rely on the estimators or experts, chances are that they can be biased, optimistic or pessimistic

2.1.3 Top-down estimation

In this technique we derive total cost from global properties using either of algorithmic or non-algorithmic technique. Then this cost is splitted to various components of the system. Top-down Estimation is more beneficial in the early stages of software development because detailed information is not available during this stage [9],[10]. Putnam's Model is an example of this technique.

Advantages:

- It requires very less detail about the project, moreover it is faster and easier to implement.
- Unlike other techniques top-down estimation focuses on activities like integration, management etc. Usually these are overlooked in other techniques.

Disadvantages:

- This technique does not take into consideration low level problems which are difficult and can increase the cost of the system.

2.1.4 Bottom-up estimation

Bottom-up estimation is opposite of Top-down estimation method. In this method we derive cost of each software component and then the result is combined to achieve the overall cost of the software. Goal is to derive system estimate from the accumulated estimate of the small component [10].

Advantages:

- This technique is more stable .

Disadvantages:

- It does not take into account the system level activities like documentation, integration and their associated costs.
- It takes more time.

2.1.5 Price to win estimation

Here we are focused more on the budget of customer rather than the functionality of the software. Overall software cost is agreed on the basis of an outline proposal and the development of software is restricted by that cost.

Advantages:

- Cost is estimated according to the budget of customer.

Disadvantages:

- This method may lead to delay in delivery of the software project, due to which software developers may suffer loss.

2.1.6 Artificial neural network based estimation

Artificial Neural Network based estimation models are trained by the use of historical data. They produce good results and the algorithmic parameter values are adjusted in such a way that the differences between the actual and predicted estimates are reduced.

Advantages:

- Artificial Neural Network [11] based estimation methods are consistent with unlike databases and they provide power of reasoning in estimation process.

Disadvantages:

- Large amount of training data is required
- No guidelines or instructions are provided for designing.

2.1.7 Fuzzy logic based estimation

Fuzzy Logic [12] based estimation is also called soft computing technique. Soft computing techniques are emerging software estimation techniques. Fuzzy logic has evolved as an important tool to solve such problems, for which mathematical models cannot be created or we can say that it is difficult to create. Development of software is many a times characterized by parameters that exhibit fuzziness. Application of fuzzy logic in cost estimation helps in overcoming many problems which exists in the already available cost estimation techniques.

Advantages:

- No training is required also this method is more flexible.
- Provides reliable estimates.

Disadvantages:

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

2.2 Algorithmic techniques

Algorithmic methods make use of equations and mathematics to perform the process of estimation. Moreover these equations are derived from research and uses inputs like Source Lines of Code(SLOC)[13],function points, and cost drivers like risk assessments, languages, design methodology etc. Models like COCOMO (Constructive Cost Model), Putnam's Model, Function Point based models and SEER-SEM Models [14] are some of the Algorithmic models.

2.2.1 Constructive cost model

One of the popular and extensively used algorithmic model for the estimation of cost and at the same time get the schedule of a developing software was given by Barry Boehm[15],[16] and is known as the Constructive Cost Model(COCOMO)[15]. The parameters and equations that are

used in this model are obtained through previous software projects. The size of code is usually given in KLOC(thousand lines of code) and the obtained effort is in Person Months(PM).Basically three models of COCOMO have been proposed by Boehm and these are as follows:

- Basic COCOMO – Being the first of the COCOMO set of models ,the formula used by this model is:

Effort = $a*(KLOC)^b$, here KLOC denotes the code size and the constant is represented by a and b. The value of these constants [17] depends on the type of project, that is whether its organic, semi-detached or embedded.

- Intermediate COCOMO – In this we obtain nominal effort estimation and the value of constants a and b differs from that of basic COCOMO. Formula used in this model is:

Effort = $a*(KLOC)^b*$ EAF. Here the effort adjustment factor is represented by EAF.

- Detailed COCOMO – This works on each sub-system separately and serves as a boon for large systems made up of non-homogenous subsystems.

Constructive Cost Models presumes the system and software requirements to be stable and predefined. But usually this situation is not always valid. This model provides some advantages but it also has some disadvantages.

Advantages:

- Simple to estimate cost

Disadvantages:

- Because estimation in COCOMO Model is done at early stages of software development, many a times it may lead to estimation failures.

As a result of these problems the newest version of COCOMO which is COCOMO II was developed in 1990 and uses broader set of data. It uses source lines of code, function point and object points as inputs. It also includes some modifications to the effort multiplier cost drivers of previous COCOMO. The obtained output is in the form of size and effort estimates later developed into a project schedule.

Advantages:

- COCOMO II proves to be an industry standard model.
- It has a clear and effective calibration process.

Disadvantages:

- Calculation of duration for small projects is unreasonable.

2.2.2 Putnam's model

This model examines many software projects and depends on distribution of manpower. The equation used in this model is as given below:

$S = E * Effort^{1/3} * t_d^{4/3}$.

Here, S is the size of the software [18] in LOC (Lines of code), Environment factor is denoted by E which depicts the development capability, t_d denotes the software delivery time and effort is represented in person year**.** Apart from this one more relation was found by Putnam which is given below:

$Effort = D_0 * t_d^3$.

Here, D_0 which represents manpower build-up factor ranges from 8 to 27 for new and rebuilt software respectively. Tool that is used for estimation of cost and manpower scheduling is SLIM and it is based on the Putnam's Model.

Advantages:

- This model is basically based on two variables which are time and size.

Disadvantages:

- Putnam's Model does not take into account other aspects of software development life cycle.

2.2.3 Function point based analysis

Function Point Analysis was developed by Albrecht [19] to measure the functionality of software projects. This method measures the size of the software, it considers internal logical files [20], external interface file, external input-output, and external inquiries from functional viewpoint metric. ESTIMACS and SPQR/20 are the models based on Function point analysis estimation.

Advantages:

- They are not dependent on the language, tools and methods of implementation.
- Development costs can be estimated in early phases of software development.
- Results are better than SLOC (Source Lines of code).

Disadvantages:

- It requires manual work which is more time consuming.
- Difficult for a new developer to estimate size of the software[21] as function point usage requires experience.

3. SELECTION OF ESTIMATION TECHNIQUES

It is quite clear from the above mentioned comparison of the techniques that as such there is not any single technique which can be credited as the best one. The merits and demerits of each technique of estimation are correlated, so an amalgamation of these techniques [22] can help in ruling out weaknesses of any particular method. It can reduce the negative effects of a technique and can help in augmentation of their individual strength. Also we can cross check one method with another. Usually it is recommended to use non algorithmic methods like estimation by analogy or expert judgement method for the projects which are known. On the other hand for larger and less known projects it proves better to use algorithmic methods. Amongst the algorithmic models, COCOMO II is much better than COCOMO I as it is not only confined to use Source lines of code but can make use of function point, object point as software metrics to measure the size of the software projects. So efforts should be made to use combination of the techniques to arrive at a better estimate of the software.

4. CONCLUSION

Predicting actual cost estimate required to develop particular software is a tedious task. Planning and budgeting of software project is largely affected by cost estimation, thus it is an essential process in software estimation. Cost estimation if done before the initiation of a project can aid in determining the features which can be included within the limited resources of the project. It also helps in reducing risks. So we can say that overall cost estimation is very impactful for the

life and schedule of a software project. Our aim should be to produce such software which are both cheap and offer a good quality. There are many methods of estimating cost but as it is clear that we cannot consider any single technique to be the best one as each of the techniques have their own advantages and disadvantages. Efforts should be made to use a combination of the estimation techniques to arrive at a better cost and quality estimate. To produce reliable estimates it is needed to have proper knowledge and understanding of each technique and the relationship between the software attributes of each.

5. REFERENCES

[1] Pressman, Roger S."Software Engineering: A Practitioner's Approach",6th Edition,McGraw- Hill, New York,USA,ISBN :13:9780073019338,2005.

[2] Khatibi Bardsiri,V.,D.N.A.Jawawi,S.Z.M Hashim,and E.Khatibi,"Increasing the accuracy of software development effort estimation using projects clustering",IET Software,2012.

[3] Patil,Lalit V.,Rina M.Waghmode, S.D.Joshi,and V.Khanna, "Generic model of software cost estimation:A hybrid approach",2014 IEEE International Advance Computing Conference (IACC),2014.

[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] Keung.J.W.,B.A. Kitchenham,et al."Analogy-X: Providing Statistical Inference to Analogy based Software Cost Estimation". Software Engineering, IEEE Transaction on 34(40:471-484,2008.

[7] M.Jorgensena, "Review of studies on Expert Estimation of Software Development Effort", Journal of Systems and Software,70(1-2):37-60,2004.

[8] Caper Jones, "Estimating Software Cost", Tata Mc–Graw Hill Edition,2007.

[9] Kusuma Kumari B.M, "Software Cost Estimation Techniques", International Journal of Engineering Research in Management and Technology, Volume -3, Issue- 4,2014.

[10] Hareton Leung,Zhang Fan , "Software Cost Estimation", Article 2001.

[11] 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) Volume: 3, Page(s): V3-487 – V3-491,2010.

[12] Attarzadeh , I. Siew Hock Ow , "Improving the accuracy of software cost estimation model based on a new fuzzy logic model" , World Applied Science Journal 8(2):117-184,2010.

[13] Albrecht A.J. and J.E. Gaffney, "Software function,source lines of codes, and development effort prediction : a software science validation ", IEEE Trans Software Engg,SE, pp.639-648,1983.

[14] Fischman ,L,K.McRitchie , and D.D.Galorath."Inside SEER-SEM", Cross Talk, The Journal of Defense Software Engineering,2005.

[15] Shruti Jain,"Survey of Various Cost Estimation Techniques", International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), Volume-1,Issue-7,September,2012.

[16] R.K.D.Black , R.P. Curnow, R.Katz and M.D.Gray, BCS Software Production Data, Final Technical Report,RADC-TR-77-116,Boeing Computer Services, Inc, March 1977.

[17] Khalifelu,Zeynab Abbasi, and Farhad Solemanian Gharehchopogh,"Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation",Procedia Technology,2012.

[18] Dizaji ,Zahera Ashegi and Khalilpour,Kamal,"Particle swarm optimization and chaos theory based approach for software cost estimation",International Journal of Academic Research,2014

[19] Yinhuan,Z., W.Beizhan,et al. "Estimation of software projects efforts based on function point",Computer Science & Education. ICCSE ,4th International Conference,2009.

[20] Mustafa,K.K Gowthaman, and R.A.Khan,"Measuring the Function Points for Migration Project: A Case Study", American Journal of Applied Sciences,2005.

[21] "FAHSCEP: Fuzzy and Analogy based Hybrid Software Cost Estimation Process", International Review on Computers and Software,2013.

[22] M.V.Deshpande, S.G.Bhirud, "Analysis of Combining Software Estimation Techniques", International Journal of Computer Application (0975-8887)", August,2010.