

# Analogy-Based Estimation with Three Point Technique

Chetan Nagar<sup>1</sup> | Dr. Asheesh Shah<sup>2</sup>

<sup>1</sup> CSE Research Scholar, Mewar University, Chittorgarh, Raj, India

<sup>2</sup> Pro President, Mewar University, Chittorgarh, Raj, India

## To Cite this Article

Chetan Nagar and Dr. Asheesh Shah, "Analogy-Based Estimation with Three Point Technique", *International Journal for Modern Trends in Science and Technology*, Vol. 03, Issue 02, 2017, pp. 05- 09.

## ABSTRACT

In the era of nineties most of the software are made for single computer and they are simple software. Today's software's is made to run on web and cloud, so it is obvious that software complexity has increased and various surveys have found that the success rate of a software project is very low, approximate 30% software becomes successful. There are many reason for software failure, effort estimation is one of them. Many methods and models are available for effort estimation, but software industries are still searching for a model of effort estimation. Analogy based estimation is one of the simple and popular methods of estimation. In this paper, we are using analogy based estimation with PERT Three Point Technique. It will increase the accuracy of analogy based estimation. Because to use Three Point Technique we do analogy three times with a different thinking. We know that in effort estimation we need to predict so many parameters, for some parameter we have a clear value but in the case of some parameter we have a doubt what should be the values and it vary according to the condition. If we know everything about the parameter then it has the least value (Best Case), If we know something about the parameter then it has average values (Average Case) and if we have a very less information about the parameter then it has worse value (Worst Case). But in estimation we cannot keep all these values, in the analogy, we have to take any one value it may be best, average or worst values. Three-point techniques with Analogy-Based Estimation allow us make estimation with three different values then take the average and it would be most correct estimation.

**KEYWORDS:** PERT (Program Evaluation and Review Technique).

Copyright © 2017 *International Journal for Modern Trends in Science and Technology*  
All rights reserved.

## I. INTRODUCTION

Software effort estimation is an mandatory and important activity of software development, it completed in the analysis phase of software development. So many method and models are invented by the researcher. But none of the models is suitable for all the industries and all the types of project and this thing insist researcher to make a new model or method which can full fill this condition. Analogy based estimation is a suitable alternative to other conventional estimation techniques. Analogy-Based Estimation (ABE) has earned good popularity within software engineering

research community because of its outstanding performance in prediction the effort with the help of old projects [1, 15]. Actually, the analogy is not only used in efforts estimation but it is used in our daily life also to solve many problems. In this approach, we find the solution of a problem on the basis of an old solution. So can say the analogy is not a new reasoning paradigm as it has been extensively studied and discussed by philosophers and scientists from thousands of years.

Software effort estimation is a process of predicting the effort required to develop a software project. While the software effort estimation may be simple in concept, it is difficult and complex in

reality [1]. From last three decades, the importance of software effort estimation has been increasing gradually. Software effort estimation is allowing us to calculate how long and how many people are required to complete a software project. It's not only estimate how many resource required for how long time, but it also provide the base for the planning and scheduling activity.

The estimation process includes size estimation, effort estimation, developing initial project schedules and finally estimating the overall cost of the project. Software development has become an essential question [2] because many projects are still not completed on schedule, with under or over estimation of efforts leading to their own particular problems [3].

As we know that we have very low percentage of successful software. Which project can be called as successful software? So the answer is: software which is completed on time, within the budget and satisfies all the requirement of customer. Accurate effort estimation is one of the major requirement of completing the project on time and within the budget. It also provides the basis for other software development activities like planning, Scheduling, and resource estimation.

## II. METHODS OF ESTIMATION

No one method is necessarily better or worse than the other, in fact, their strengths and weaknesses are often complimentary to each other. To understand their strengths and weaknesses is very important when you want to estimate our projects [4].

Software industries are using two types of effort Models or methods:

**Algorithmic Model:** - This software cost estimation technique use the mathematical equations to perform the software estimation. The mathematical equations are based on historical data or theory. SLOC (source line of code), function points, and other cost drivers are the inputs. For most algorithmic model, calibration to the specific software environment can be performed to improve the estimation. 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. COCOMO, Function Point Based Estimation, UseCase Point Estimation, SLIM, SEER SEM, SASET and RAVIC are the example of an algorithmic method of estimation.

**Non-Algorithmic Model:** - In the non-algorithmic

model, the estimation can be done by using the previous projects previous experiences which are similar to the project under estimation.

1. **Expert judgment:** Expert judgment techniques involve consulting with software cost estimation expert or a group of the experts to use their experience and understanding of the proposed project to arrive at an estimate of its cost. It is the most usable methods for the software cost estimation. Mostly companies used this method for generating the cost of the product. Generally speaking, a group consensus technique, Delphi technique, is the best way to be used. [5].
2. **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 splitter 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 [6], [7]. Putnam's Model is an example of this technique. This method 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.
3. **Bottom-Up Estimation:** Expert judgment techniques involve consulting with software cost estimation expert or a group of the experts to use their experience and understanding of the proposed project to arrive at an estimate of its cost. It is the most usable methods for the software cost estimation. Mostly companies used this method for generating the cost of the product. Generally speaking, a group consensus technique, Delphi technique, is the best way to be used. [7].
4. **Price-to-Win:** 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. Cost is estimated according to the budget of the customer. This method may lead to delay in delivery of the software project, due to which software developers may suffer loss.
5. **Analogy-Based Estimation:** used. [5, 6] Effort estimating by analogy means comparing the proposed project to previously completed



similar project where the project development information is known. Actual data from the completed projects are extrapolated to effort estimate the proposed project. Analogy method can be used either at the system level or at the component level [9]. This method using following estimating steps:

1. Find out the necessary characteristics of the proposed project.
  2. Choose the most similar completed projects whose characteristics have been stored in the historical database.
  3. Find the estimate for the proposed project from the most similar completed project by analogy.
6. **PERT Three Point Estimate Technique:** The PERT Three Point Estimate technique is a type of three point estimate. The only difference is that it applies weighting so that the most-likely estimate is weighted 4 times more than the other two estimates (optimistic and pessimistic). This formula is most valuable in estimating time or cost of activities for projects that are especially unique, such as in research and development where there are many unknowns. For projects that are similar to previous projects and there is good historical data and expert experience, the formula is less useful because you could use other techniques like analogous estimating (based on previous experience and projects). Pert Estimate

$$E = (o + 4m + p) / 6$$

Where  $E$  is Estimate;  $o$  = optimistic estimate;  $p$  = pessimistic estimate;  $m$  = most likely estimate.

### III. SUGGESTED METHOD

Analogy-Based Estimation generates a single value, which is the estimated effort based on previous similar project. Many time during the prediction we thought that what should be the value of this parameter because its value depends on the condition. For some parameter we have a clear value but in the case of some parameter we have a doubt what should be the values and it vary according to the condition. If everything goes wright then it has least value (Best Case), if something goes wrong then it has average values (Average Case) and if everything goes wrong then it has worse value (Worst Case). But in estimation we cannot keep all these values, in analogy, we have to take any one value it may be best, average or worst

values. Three-point techniques with Analogy-Based Estimation allow it to take the average and put a correct value.

It is a question that, in analogy based estimation we estimated the effort on the basis of any old project, so how we can incorporate or use Three Point Technique with analogy based estimation. In Analogy based estimation it rarely happens that we have the same project in our database for the analogy, if we find the same project, it also needs some prediction because we cannot put the same values because now time, technology and many other parameters are changed. When we have a more similar project (70% or 80% or more similar), it can be called more analogy less prediction, But when we have not found similar project or less similar (50% or 60% Similar), it can be called less analogy more prediction. So here Three Point Technique will work to make estimation more accurate.

In analogy based estimation, most of the time we compare the project with their functionality. Three Point Technique can be used in two way one at parameter level and second at module or project level.

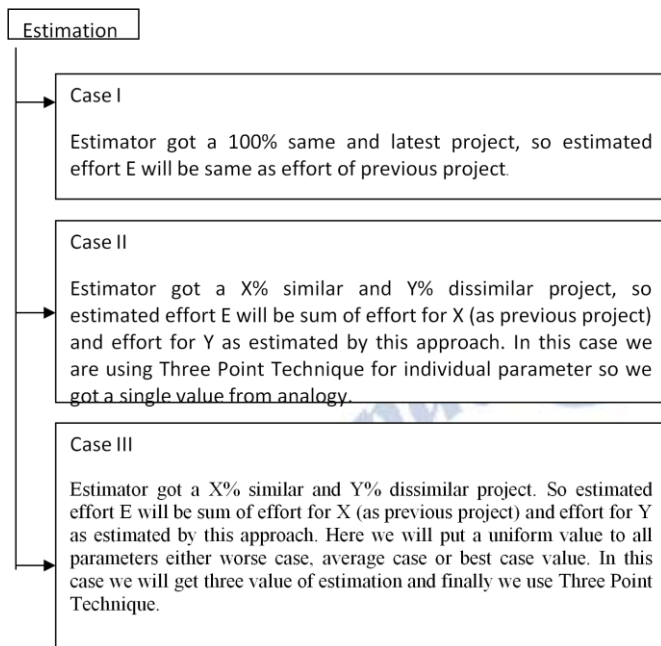
For any parameter if we are getting three values (Best, worse and average) then we have two ways to use Three Point Technique:

1. We can every time use Three Point Technique for individual value. In this case will get a single value as the total of estimated values calculated through Analogy Based Estimation.
2. In second way we will put a uniform value to all parameters either worse case, average case or best case value. In this case we will get three value of estimation and finally we use Three Point Technique to get an average value.

### IV. RESULT

In the study of Three Point Technique, we have found that this technique is used when we have less experience or less previous project data or inexperienced team. But in our research, we have applied this method to analogy based estimation to improve the accuracy or results of analogy based estimation. Because it is rare that we find the same project for the analogy, most of the time we have got the approximate similar project. It means we need to predict efforts for new part of the project.

In the below diagram we have explained how this method will give the result and what are the different cases.



**Fig1: Analogy Based Estimation with Three Point Technique.**

It is concept that we have describe in this paper. To test this concept we can take some old project and estimate the effort. For testing purpose we have explain this concept in front of small software development company, they work on the concept and draw the conclusion is that, they are getting a improvement of 5-10%, but they are not ready to share the details due to their company plicies.

It may happen that sometimes we may more optimistic due to over confidence or sometimes we may more pessimistic due to fear. Three Point Technique allow us to adjust these values. It finds an average of best case, average case and best case values. Three Points will minimize our mistake whether optimistic or pessimistic.

## V. CONCLUSION

Selection of a model for estimation is really a difficult task. There is no single method which is suitable for all types of projects. Analogy based estimation is one of the simple and most favorite method of estimation. In this paper, we have added Three Point technique to improve the results of estimation. We can use three point techniques in two ways which we have shown in result section. It does not need to say that Analogy-Based Estimation is not most suited for all types of project, but it is a batter alternative to algorithmic model. Several studies suggest that analogy based estimation is most suitable in some particular situation. Analogy based estimation has improve the estimation and in this paper we have used

Three Point Technique which improves the Analogy Based Estimation result. Although it is not compulsory that Three Point can only be used with Analogy-Based Estimation, it can also be used with some other algorithmic or non-algorithmic method also. But in this paper, we have chosen Analogy based estimation.

In our previous research papers, we have already suggested that: 1. no method is suitable for all projects and all software industries each have their own problem. So we have to make a localized version of model or method. 2. We must use a monitoring policy and 3. Multistage estimation can make our estimation more accurate.

## REFERENCES

- [1] Maged A. Yahya, Rodina Ahmad, Sai Peck Lee, "Effects of Software Process Maturity on COCOMOII's Effort Estimation from CMMI Perspective" In 2008 IEEE, Department of Software Engineering, University of Malaya, pp. 255-256.
- [2] A.J. Albrecht and J.E. Gaffney, "Software function, source lines of code, and development effort prediction: a software science validation", IEEE Transactions on Software Engineering, pp. 639-647, 1983.
- [3] B.W. Boehm, "Software Engineering Economics", Prentice- Hall, Englewood Cliffs, NJ, USA, 1981
- [4] L.H. Putnam, "A general empirical solution to the macro software sizing and estimation problem", IEEE Transactions on Software Engineering, pp. 345-361, July 1978.
- [5] M.Jorgensena, "Review of studies on Expert Estimation of Software Development Effort", Journal of Systems and Software,70(1-2):37-60,2004.
- [6] Kusuma Kumari B.M, "Software Cost Estimation Techniques", International Journal of Engineering Research in Management and Technology, Volume -3, Issue- 4,2014.
- [7] Hareton Leung,Zhang Fan , "Software Cost Estimation", Article 2001.
- [8] 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.
- [9] F. Walkerden and R. Jefferey, "An empirical study of analogy-based software effort estimation," Empirical Software Engineering, vol. 4, pp. 135-158, 1999
- [10] M. Azzeh, A replicated assessment, and comparison of adaptation techniques for analogy-based effort estimation, Journal of Empirical Software Engineering vol. 17, pp. 90-127, 2012.
- [11] M. Shepperd, C. Schofield, Estimating software project effort using analogies, Journal of IEEE Transaction on Software Engineering, vol. 23, pp. 736-743, 1997.



- [12] Z. M. Yuan, L. Zhang, G. H. Zhan, A novel web-based online examination system for computer science education, In proceeding of the 33rd Annual Frontiers in Education, 2013, S3F7-10.
- [13] WebBased online Secured Exam; B.Persis Urbana Ivy, A.shalini, A.Yamuna/International Journal of Engineering Research and Applications (IJERA) ISSN:2248-9622 www.ijera.com Vol. 2, Issue 1, Jan-Feb 2012, pp.943-944943.
- [14] Online Descriptive Examination and Assessment System.L. Zhang, et al., Development of Standard Examination System of Special Course for Remote Education, Journal of Donghua University (English Edition), 2013, Vol. 19, NO.1, 99-102.
- [15] Challenges of Online Exam, Performances and problems for Online University Exam; IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 1, January 2013 ISSN (Print): 1694-0784 — ISSN (Online): 1694-0814. [www.IJCSI.org](http://www.IJCSI.org)
- [16] Al-Mashaqbeh, I.F. Al Hamad, A. Student's Perception of an Online Exam within the Decision Support System Course at Al alBayt University Conference publication Pages: 131135 7-10 May 2010..
- [17] Design and Development of the Online Examination System based on B/S Structure. HongmeiNieMath, Physics and Information Engineering College Zhejiang Normal University Jinhua, China E-mail: [nhm@zjnu.cn](mailto:nhm@zjnu.cn)
- [18] Shuffling Algorithms for Automatic Generator Question Paper. Nor ShahidabtMohdJamail Abu BakarMd Sultan Faculty of Computer Science and Information Technology Universiti Putra Malaysia, 43400 UPM SERDANG, Selangor, Malaysia. E-mail: shahidajamail@yahoo:com
- [19] Hanxiao Shi, Guodong Zhou and PeideQian (2010), An A ttribute - based Sentiment Analysis System, Information Technology Journal, pp 1607 - 1614. PapriChakraborty (2012), Developing an Intelligent Tutoring System for Assessing Students Cognition and Evaluating Descriptive Type Answer, IJ MER, pp 985 - 990.