

# Survey on Effective Software Effort Estimation Techniques

Rathi. J<sup>1</sup>, Kamalraj. R<sup>2</sup>, Karthik. S<sup>3</sup>

**Abstract**— Software effort estimation is a very important task in the software engineering field. In this paper gives the overview about the software effort estimation techniques. The techniques are expert estimation, parametric model, algorithmic model, COCOMO, gray relational analysis with genetic algorithm, Ridge Regression (RiR), Multivariate Adaptive Regression Splines (MARS), Robust Regression (RoR), Classification and Regression Tree (CART), Case Based Reasoning (CBR), analogy weights with genetic algorithm are available in the field. And this paper concludes that we cannot say a particular technique is best fit for all the situations to give an accurate estimation. So that we make a careful comparison between all estimation approaches and choose the appropriate technique for each task. It will help us to make accurate software effort estimation by these estimation techniques.

**Index Terms**— Software Effort Estimation, Software Cost Estimation, Similarity Measures, Historical Projects.

## 1. Introduction

Estimation is a frequently occurring task in our life. For example if we leave from one place to another place we should calculate the time to reach. And the estimation is based on the historic data or the past experience. The estimating parameters are varying from the different tasks. Cost, resources, manpower, technical equipment, time, schedule and other similarities between the projects are the parameters of the estimation. The present life style software products are very important in every once life. So the software industry is looking to produce quality products with low costs. Software engineering is complex task to define its exact scope. Before stating the project in the cost estimation is very challenging to the project managers. So we need the accurate time, cost and the technical resources to be use to build the software. So that to estimate the effort to develop software there are more techniques are available in the field. As per the situation need to choose the appropriate estimate technique. If we made inaccurate estimation the company and the customer both are suffered by the time or cost. So that the project managers should give the accurate estimation to develop a software. And the software industries have a very big competition to get the market value in field. So they should provide good service to the people to retain the market. The software effort estimation is the process of predict the effort require to develop the project. Accurate software cost estimations are very tough to both project developers and client. The project managers used to do the request for

proposals from the client, negotiating for contract, scheduling the project delivery, monitoring and control. The accurate estimation avoids the project is over running the actual schedule. In this paper describes the several methods behind the software effort estimation. Several reasons imposed to give dissatisfactory about the software against the actual requirements. Those are ignoring past/historic data, uncertainty of requirements, non-estimation, management pressure, unskilled estimators and budget constraints. Yardstick to measure the size of the code is Lines of Code (LOC) and Function analyze is used ton met Point(FP). To produce the accurate effort estimation the four metrics are used. The past project execution metric is used to analyze and access the capabilities, strengths and weakness of the domain, process and technology skills. Past experience gives the staring and finishing dates. Present project execution metric gives the measuring the ongoing projects. Future metric is used to set the targets to complete a task or to deliver a software artifact.

### 1.1 Importance of accurate effort estimation

- To determine what resources to commit to the project and how well these resources will be used.
- To assess the impact of changes and support re-planning.
- To manage and control when resources are better matched to real needs.
- To produce a line function.
- To give the effort in the basis of time, cost and technical resource.

### 1.2 Cost factors

- Reliability of the Software, Database Size used for the project and Product Complexity are based on the product.
- Execution time and main memory constraint, Virtual Memory and Computer turn around time are based on the computer.
- Analyst capability, Application experience, Programmer capability and language are based on the personal.
- Project size, Complexity, Languages and Skill levels are the data inputs for the estimation.

### 1.3 Impact of Inaccurate Estimations

- The project may be failed and customer dissatisfaction.
- Poor estimation causes loss of profit for the company.
- Inability to anticipate skills of project team members and overlooked tasks.
- Lack of coordination of systems development, technical services, operations, data administration functions during development.

### 1.4 Ingredients of a good estimation

- Activity Scope: The size of the software in terms of functionality that it delivers or the terms of lines of code are the scope of a software development.

---

*Manuscript received Oct 15, 2011.*

<sup>1</sup>Rathi.J II-ME (Software Engineering) SNS College of Technology, E-Mail: rathinaidu.me@gmail.com, Coimbatore, TamilNadu.

<sup>2</sup>Kamalraj.R, AP/CSE, SNS College of Technology, E-Mail: mailtokamalraj@yahoo.com, Coimbatore, TamilNadu.

<sup>3</sup>Karthik.S, Dean/CSE, SNS College of Technology, E-Mail: profskarthik@gmail.com, Coimbatore, TamilNadu.

- Working Environment: The environment in which the activity is being executed is very important to build software.
- Consistency: The historic data is useful information to estimate the similar project which already developed. Some times new observations, issues and hurdles will be occurs. This will make the get experience to handle the situation.
- Tools: Tools that are execute an activity and play a vital role in defining the effort and the time taken to complete the activity. And be different forms, shapes and size depends on the various approaches.
- Learning form past experience: the people who learned lessons from the past mistakes. So we cannot learn everything from the past experience but also learn from well defined process recorded experience and mistakes to give a correct solution.

## 2. Software Effort Estimation Techniques

### 2.1 Expert Estimation

The expert estimation is the judgmental process extract the information from the experts who got experience in the field. The Delphi Method is an expert based estimation. A well formed communication technique, originally developed as a systematic, interact with the panel of experts. The Planning Poker is to avoid the influence of the other participants. If a number is spoken, they give a suggestion. It forces people to think independently and propose their numbers simultaneously. This planning poker and the Delphi method based on the Group Discussions. These methods are used in the Agile software development.

### 2.1 Analogy Based Software Estimation

The basic idea of Estimation by Analogy is to create accurate estimates for new projects by comparing the new project to similar projects from the past. The goal is to utilize the known effort estimation of the completed projects and using it to estimate the effort for the new project. Before starting with estimation by analogy we have to determine how a project can be described like type of application domain, number of inputs given by the users and number of output screens will be shown. We have to be sure that at least one variable can act as a size driver. The most important decision to make when estimating by analogy is to determine similarity and how the known effort values are used to make estimations for a new project. Measuring Analogy is done by tools. The Analogy Software Tool (ANGEL) is a tool to automate the analogy measurement process. The prototype defines that optimum combination of variables for finding analogies. Templates are used to record project data which can be configured for the data collection of an organization. It tracks the project status and actual effort and records it. In analogy based effort estimation used to improve the accuracy for software effort estimation with the data mining techniques which are used to find the similarities. The optimal adjustment algorithm is more suitable for the analogy based effort estimation [2].

### 2.2 Parametric models

Use effort drivers representing characteristics of the target system and the implementation environment used to predict the new effort. In parametric model the top-down approach is used to produce overall estimation using effort driver.

And bottom-up approach is no past project data is available, when we use this model. Here break project into smaller and smaller components. Estimate costs for the lowest level activities and using lowest level calculate the higher level estimation. This model base on historical data about the software project. It will find the time factors affecting the project to complete the time and effort estimation. The parameters are the personnel, programmer skill set, tools to develop a software and reuse factors.

### 2.3 COCOMO

The COCOMO model is the complete and documented model used in effort estimation. COCOMO is based on Boehm's analysis of a database of 63 software projects. The model provides detailed formulas for determining the development time and schedule, overall development effort, effort breakdown by phase, activity and maintenance. It estimates the effort in persons-months of direct labor. SLOC(Source Lines of Code) is the main factor and expressed in KDSI(Thousand Lines of Code). Classic waterfall model and good management practices with no slack time assumes of the COCOMO. This model designed by three versions is basic, intermediate and detailed. Three classes of process modeling are embedded, organic and semidetached. Embedded is characterized by tight constraints, changing environment and unfamiliar surroundings. The organic small relative to project size, team size and have a stable environment. Semidetached is the software systems falling under this category are mix of the embedded and organic. The formula for estimate the Effort =  $a \cdot KDLOC^b$ . Where a and b are two parameters of the model whose specific values are selected upon the classes of the software system.

### 2.4 Gray Relational Analysis with Genetic Algorithm

The Gray Relational Analysis used to build formal software effort estimation. In the software industry most of the projects are not completed and have unnecessary relationship between with effort drivers. Gray relational analysis integrated with genetic algorithm will give a more accurate value. Genetic algorithm used as a searching technique and help to solve the optimization problems in more areas. Here Gray means black and white. Black refers the required data is entirely unavailable. White refers the required data is entirely available. This Gray helps to bridge the gap between the black and white. When integrating the Genetic Algorithm with Gray Relational Analysis gives the better performance to compare the methods like Case Based Reasoning (CBR), Classification and Regression Trees (CART) and Artificial Neural Networks (ANN) [1].

### 2.5 Ordinary Least Square Regression (OLS)

This method is used traditionally to estimate the effort. The goal of the model is estimating the unknown parameters in a linear regression model. It minimizes the sum of squared vertical distances between the observed responses in the dataset and the responses predicted by the linear approximation. A data set is being fit to the well documented project.  $e_i$ , the dataset have dependent  $x_i(1)$  to  $x_i(n)$  Multiple independent variables and called as multiple regression.

$$e_i = X_i \beta + b_0 + \epsilon_i$$

Where  $X_i$  - Row vector contains the values of the  $i$ th observation  $x_i(1)$  to  $x_i(n)$ .

$\beta$  - column vector contains the slope parameters which is estimate by the regression model.

$b_0$  - Intercept Scalar.

$e_i$  - Error associate with each observation.

### 2.6 Robust Regression (RoR)

Iteratively Reweighed Least Square (IRLS) is application of the Robust Regression (RoR). This is updated version of the OLS Regression. And it is less threats to the existence of the outliers in the data set. Robust regression is a form of regression analysis designed to overcome some limitations of traditional parametric and non-parametric methods. Regression analysis makes to find the relationship between one or more independent variables and a dependent variable.

### 2.7 Ridge Regression (RiR)

This regression method highly correlated attributes with attributes with the OLS helps to solve the potential problems. As per the BLUE the  $X'X$  should be a non singular. But some cases these are found singular. So that the ridge parameter will be used.

### 2.8 Least Median of Square Regression

LMS uses the break point concepts. If the break point is fifty percentage, this LMS will be the alternative to the RoR.  $K^*$  is the minimum percentage of incorrect data LMS function =  $\min \text{median}(e_i)$ . where  $e_i$  error associated with  $i$ th observation. And this technique is not suitable for all situations for the estimation.

### 2.9 Multivariate Adaptive Regression Splines (MARS)

MARS is a non-linear and non-parametric techniques. This has different properties like interpretability, capability of modeling complex non-linear relationship and fast growth model construction.

### 2.10 Classification and Regression Tree (CART)

This CART method is used the well defined decision trees and it constructs the algorithm model. This will form a binary tree recursively dividing the data into data set and it will perform the stop criteria will meet.

### 2.11 Case Based Reasoning (CBR)

Case based reasoning is the process of finding the most similar projects. Its works like the expert analysis. Here the most similar distances are calculated by the distance.

### 2.12 Analogy weights with genetic algorithm

Effort estimation accuracy is integrating genetic algorithm to determine the proper weighted similarity measures and non-linear weighted methods. The effort drivers play a vital role and it will be construct as the software requirements. The effort drivers must select the suitable task for the estimation. Un weighted methods also to be used equally and to measure the effort driver. Dissimilar effort method is used the un weighted method[4].

## 3. Conclusion

In this paper presented an overview of the various techniques currently available for software effort estimation in the software industry. Software effort estimation is a very important task in the software engineering field because the future of the project depends on the estimation report. The techniques discussed about expert estimation, parametric model, algorithmic model, COCOMO, gray relational analysis with genetic algorithm, Ridge Regression (RiR), Multivariate Adaptive Regression Splines (MARS), Robust Regression (RoR), Classification and Regression Tree (CART), Case Based Reasoning (CBR), analogy weights with genetic algorithm. According to the empirical study we cannot say this technique is best fit for all the situations to give accurate estimation. So that we made comparison between all estimation approaches and the analogy based software effort estimation will give accurate results to compare others. Within that data mining techniques are applied with analogy techniques to get the accurate data set from large dataset and it will give a better results.

## 4. References

- [1]Sun-Jen Huang, Nan-Hsing Chiu and Li-Wei, "Integration of Gray Relational Analysis with genetic algorithm for software effort estimation" ELSEVIER Aug2007.
- [2] Nan-Hsing Chinu and Sun-Jen Huang, "The Adjusted Analogy Based Software Effort Estimation Based on Similarity Distances" ELSEVIER 2006 July.
- [3]Andrea De Lucia, Eugenio Pompell and Silvio Stefanucci, "Assessing effort estimation models for corrective maintenance through empirical studies" ELSEVIER July 2004.
- [4]Sun-Jen Huang and Nan-Hsing Chiu, "Optimization of analogy weights by genetic algorithm for software effort estimation" ELSEVIER February2006.
- [5]Karel Dejaeger, Wouter Verbeke, David Martens and Bart Baesens, "Data Mining Techniques for Software EffortEstimation: A Comparative Study" IEEE Trans on Software Engg, Vol. 38, No. 2, March/April 2012.
- [6]Stefan Koch and Johann Mitlohner " Software project effort estimation with voting rules", ELSEVIER December 2008.
- [7]Mahmoud O. Elish,"Improved estimation of software project effort using multiple additive regression trees" ELSEVIER 2009.
- [8] M. Jørgensen and M. Shepperd, "A Systematic Review of Software Development Cost Estimation Studies," IEEE Trans. Software Engg., vol. 33, no. 1, Jan 2007.
- [9] Tim Menzies, Zhihao Chen, Jairus Hihn, and Karen Lum, "Selecting Best Practices for Effort Estimation" IEEE Trans on software engg vol. 32, No. 11, Nov 2006.
- [10] Magne Jorgensen and Stein Grimstad, "The Impact of Irrelevant and Misleading Information on Software Development Effort Estimates: A Randomized Controlled Field Experiment" IEEE Transaction, vol 37, No.5, Sep/Oct 2011.
- [11] Jacky Wai Keung, Barbara A. Kitchenham and David Ross Jeffery, "Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation". IEEE Trans on Software Engineering Vol. 34, No. 4, Jul/Aug 2008.

[12] Gan Wang, Ricardo Valerdi and Jared Fortune, "Reuse in Systems Engineering" IEEE Systems Journal, Vol. 4, No. 3, Sep2010.

[13] Parthasarathi, "Practical Software Estimation" Person Education infosys.

[14] <http://en.wikipedia.org>

[15] Ning Nan and Donald E. Harter, "Impact of Budget and Schedule Pressure on Software Development Cycle Time and Effort" IEEE Trans on Software Engineering, Vol. 35, NO. 5, Sep/Oct 2009.

[16] Heejun Park and Seung Baek, "An empirical validation of a neural network model for software effort estimation" ELSEVIER 2007.

[17] Barbara A. Kitchenham, Emilia Mendes, and Guilherme H. Travassos, "Cross versus Within-Company Cost Estimation Studies: A Systematic Review" IEEE Trans on Software Engg, Vol. 33, No. 5, May 2007.

[18] Martin Auer, Adam Trendowicz, Bernhard Graser, Ernst Haunschmid, and Stefan Biffl, "Optimal Project Feature Weights in Analogy-Based Cost Estimation: Improvement and Limitations" IEEE Trans on Software Engg, Vol. 32, NO. 2, Feb 2006.

[19] Eugenio Capra, Chiara Francalanci and Francesco Merlo, "An Empirical Study on the Relationship among Software Design Quality, Development Effort, and Governance in Open Source Projects" IEEE Trans on Software Engg, Vol. 34, No. 6, Nov/Dec 2008.

[20] Randy K. Smith, Joanne E. Hale and Allen S. Parrish "An Empirical Study Using Task Assignment Patterns to Improve the Accuracy of Software Effort Estimation" IEEE Trans on Software Engg, Vol. 27, No. 3, March 2001.

[21] Kardile Vilas Vasantrao "Understanding of Software effort Estimation at the early Software Development of the life cycle - A literature View" IJERAVol. 2, Issue 1, Jan-Feb 2012, pp.848-852.

## 5. Biography



J. Rathi received M.Sc (Software Engineering) from Bannari Amman Institute of Technology affiliated to Bharathiar University. She was working as a lecturer in the Department of Computer application from Bannari Amman Institute of Technology, Sathyamangalam. And doing her Master of Engineering (Software Engineering) in SNS College of Technology affiliated to Anna University Chennai. She is an active member in CSI Student Chapter. Her research includes Software Cost Estimation, Software Engineering, Software Testing, Data Mining and Cloud Computing.



R. Kamalraj received his B.E. degree in Computer Science & Engineering from Bharathiyar University, Coimbatore, Tamil Nadu, INDIA in 2002, the M.E. degree in Computer Science & Engineering from Anna University, Chennai, Tamil Nadu, INDIA in 2009, and pursuing Ph.D. degree in

Software Testing and Quality Management at Anna University of Technology, Coimbatore. He has published 7 papers in international journals and 1 paper in National Journal. He is having 9 years of teaching experience in 4 different engineering colleges. At present he is working as an Assistant Professor in the Department of Computer Science and Engineering at SNS College of Technology, Coimbatore. His research interests include Software Testing, Software Quality Management and Data Mining.



Professor Dr.S.Karthik is presently Professor & Dean in the Department of Computer Science & Engineering, SNS College of Technology, affiliated to Anna University- Coimbatore, Tamilnadu, India. He received the M.E degree from the Anna University Chennai and Ph.D degree from Ann University of Technology, Coimbatore. His research interests include network security, web services and wireless systems. In particular, he is currently working in a research group developing new Internet security architectures and active defense systems against DDoS attacks. Dr.S.Karthik published more than 35 papers in refereed international journals and 25 papers in conferences and has been involved many international conferences as Technical Chair and tutorial presenter. He is an active member of IEEE, ISTE, IAENG, IACSIT and Indian Computer Society.