# Prediction of Software Project Effort Using Linear Regression Model

Mohd Sadiq, Aleem Ali, Syed Uvaid Ullah, Shadab Khan, and Qamar Alam

*Abstract*—**It is well known fact that predicting software effort for the software development projects with any acceptable degree remains challenging. In this paper we have used the organic software projects because in each case the projects size lies between 2-50 KLOC. In this paper we have applied the linear regression model i.e. Effort = -1.5 + 0.1804 FP to predict the software project effort and function point. After obtaining the software effort, project manager can arrange the project progress, control the cost and ensures the quality more accurately.**

*Index Terms*—**Software effort, function point, MRE.**

## I. Introduction

Software development effort estimation is a branch of forecasting that has received increased interest in academia, application domains and media. Efficient development of software requires accurate estimates. Unfortunately, software development effort estimates are notorious for being too optimistic. Inaccurate software estimates causes trouble in business processes related to software development such as project feasibility analyses, budgeting and planning. It is unrealistic to expect very accurate effort estimates of software development effort because of the inherent uncertainty in software development projects, and the complex and dynamic interaction of factors that impact software development efforts. Still, it is likely that estimates can be improved because software development effort estimates are systematically overoptimistic and very inconsistent. Even small improvements will be valuable because of the large scale of software development. Software researchers have addressed the problems of effort estimation for software development projects since at least the 1960s. Research is found in areas such as:

1) Creation and evaluation of estimation methods. Describes work on the creation and evaluation of estimation methods, such as methods based on expert judgment, structured group processes, regression-based models, simulations and neural networks.
2) Calibration of estimation models. Tailoring a model to a particular context (calibration) has been found to be difficult in practice. Problematic issues are related to, among others, when, how and how much local calibration of the models that are beneficial.
3) Software system size measures. The main input to estimation models is the size of the software to be developed. It has been proposed many size measures, for example based on the amount of functionality that is described in the requirement specification.
4) Uncertainty assessments. Software developers are typically over-confident in the accuracy of their effort estimates. Realistic uncertainty assessments are important in order to enable proper software project budgets and plans.
5) Measurement and analysis of estimation error. Proper accuracy measurement is essential when evaluating estimation methods, and identifying causes of estimation error.
6) Organizational issues related to estimation. Organizational issues such as processes to control the cost and scope of the project may have a large impact on estimation accuracy.
7) Measuring and analyzing estimation error are the basis of estimation learning related activities, such as deciding whether or not an organization has an estimation problem, identifying risk factors related to project performance in software development, and, evaluating and improving estimation and uncertainty assessment methods and tools.

The most commonly used measure of estimation error is the Magnitude of Relative Error (MRE). The mean MRE (MMRE) is often used to average estimation error for multiple observations. It is not unproblematic to use MMRE as a measure of estimation accuracy, and several other measures, such as PRED and MER, is sometimes used. However, all estimation error measures have shortcomings. Hence, the measure that should be used in any given case depends on the context.

The rest of the paper is organized as follows: In section 2 we have explained all the background and related work that are based on the prediction of the effort from the linear regression model and also from other techniques. Brief description about the software project effort estimation and function point analysis are given in section 3, and section 4 respectively. Section 5 contains the case study and experimental work. Finally we conclude the paper in section 6.

Manuscript received August 15, 2012; revised October 12, 2012.

Mohd Sadiq and Aleem Ali are with Section of Computer Engineering, University Polytechnic, Faculty of Engineering and Technology, Jamia Millia Islamia (A Central University), New Delhi-110025, India (e-mail: sadiq.jmi@gmail.com)

Syed Uvaid Ullah is with the Department of Electrical and Electronics Engineering, Patel Institute of Technology (RGPV), Bhopal, India.

Shadab Khan is with the Department of Computer Science and Information Technology, Sunder Deep College of Engineering and Technology, Dasna, Ghaziabad, U.P., India, affiliated to U.P. Technical University, Lucknow, U.P, India

Qamar Alam with the AL-Falah School of Engineering and Technology, Dhauj, Faridabad, affiliated to Maharshi Dayanand University, Rohtak, Haryana, India

DOI: 10.7763/IJIEE.2013.V3.298

## II. Background and Related Work

In [1] the authors have developed a tool that are based on software engineering matrices, this tool is used to evaluate the function point of software. In [2] the authors conduct a set of machine learning experiments with software cost estimation data from two separate organizations. In this paper the first data set consists of 104 vectors and second data set consists of 434 vectors extracted from a Electronics Commerce Software and Fleet Management Software respectively. In [3], Ingunn Myrtveit et al. have worked on reliability and validity of Software Prediction Models. In this paper the authors have used 3 different measuring procedure i.e. reliability and validity; cross-validation; and accuracy indicators. These papers have used Finnish data set in order to generate the linear regression model and also log-linear regression model. In the simulation study of this paper that is based on Finnish data set and the author opted the log-linear model. In [4], Iman Attarzedh et al. proposed a New Software Cost Estimation Model based on Artificial Neural network. In [5] the effort estimation is evaluated on the basis of genetic algorithm. There are so many approaches to estimate the effort like Machine learning approach, genetic algorithm, ANN, and algorithmic approaches [6]-[14]. In [11], [12] authors have developed a tool to estimate the software risk and cost.

## III. Software Project Effort Estimation

In this paper we have included function points as an algorithmic method since they are dimensionless and therefore need to be calibrated in order to estimate the error. In the software Engineering literature there are so many models that are used to estimate the effort. Some of the important estimation techniques are [5]

1) SEL –Model
2) Walston- Felix Model
3) COCOMO basic Model
4) COCOMO Intermediate Model
5) Intermediate Organic Model

In this paper we have used the organic mode of COCOMO basic model. The general form of the Effort can be written as

$$E = a\ LOC^{b,} \tag{1}$$

where $E$ is the effort, LOC is the size typically measured in thousand lines of code or function points, $a$ is the productivity parameter and $b$ is an economic or diseconomies of scale parameter. Apart from this approach we have another technique that is based on algorithmic approach which is used to calibrate a model by estimating values for the parameters. The most straightforward method is to assume a linear model. Using regression analysis the model can be represented as:

$$E = a1 + a2\ S \tag{2}$$

where $a1$ represents fixed development cost and $a2$ represents productivity [7].

Effort estimation predicts how many hours of work and how many workers are needed to develop a project? The effort invested in a software project is probably one of the most important and most analyzed variables in recent years in the process of project management. The determination of the value of this variable when initiating software projects allows us to plan adequately any forthcoming activities. As far as estimation and prediction is concerned there is still a number of unsolved problems and errors. To obtain good results it is essential to take into consideration any previous projects.

Estimating the effort with a high grade of reliability is a problem which has not yet been solved and even the project manager has to deal with it since the beginning. Several methods have been used to analyze data, but the reference technique has always been the classic regression method. Therefore, it becomes necessary to use some other techniques that search in the space of non linear relationship. Some works in the field have built up models (through equations) according to the size, which is the factor that affects the cost (effort) of the project. The equation that relates size and effort can be adjusted due to different environmental factors such as productivity, tools, complexity of the product and other ones. The equations are usually adjusted by the analyst to fit the real data.

## IV. Function Point Analysis

Function points were defined in 1979 in A New Way of Looking at Tools by Allan Albrecht at IBM. The functional user requirements of the software are identified and each one is categorized into one of five types: outputs, inquiries, inputs, internal files, and external interfaces. Once the function is identified and categorized into a type, it is then assessed for complexity and assigned a number of function points. Each of these functional user requirements maps to an end-user business function, such as a data entry for an Input or a user query for an Inquiry. This distinction is important because it tends to make the functions measured in function points map easily into user-oriented requirements, but it also tends to hide internal functions (e.g. algorithms), which also require resources to implement. Over the years there have been different approaches proposed to deal with this perceived weakness; however there is no ISO recognized FSM Method that includes algorithmic complexity in the sizing result.

In this paper we have not defined the complexities of the projects i.e. low, average and high. These three parameters are generally used to calculate the unadjusted function point (UFP). To find out the values of the UFP five different functional units are required. In the FP literature the functional units contains external inputs, external outputs, external inquiries, internal logical files and external interface files. Finally the FP is calculated using the following relationship.

$$FP = UFP \times VAF \tag{3}$$

whereas the VAF i.e. value adjustment factor is calculated using 14 general system characteristics. To get the detailed description about the 14 general system characteristics, readers are advised please refer to [13]. The detailed description about the function point analysis and its computation is available in [1], [9], [10].

To find out the values of the FP, we have used the relationship between LOC and FP given in [13].

## V. Case Study and Experimental Work

Generation of artificial data with known properties to generate the software engineering data set modeling technique was first proposed by Pickard et al. [6]. It provides the researchers with a great deal and more control over the characteristics of a data set. With the help of the result of [6] in this paper we have considered 10 different projects of organic mode and the size of the projects lies between 2-50 KLOC. The information about the Lines of code of each project is available in Table I.

TABLE I

| Project No. | Lines of code |
|---|---|
| 1. | 14000 |
| 2. | 12000 |
| 3. | 09000 |
| 4. | 06000 |
| 5. | 08000 |
| 6. | 05000 |
| 7. | 10000 |
| 8. | 15000 |
| 9. | 11000 |
| 10. | 13000 |

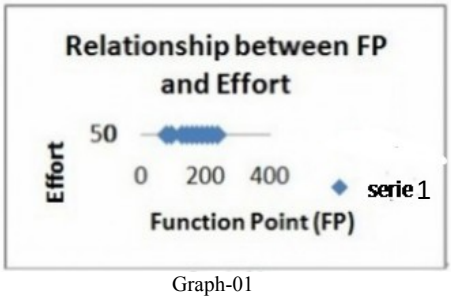
Using the results of [13] we can roughly estimate the value of the function point. In this paper we have considered that all the projects are written in C++, so in this case the value of LOC/FP=64. We have applied the basic COCOMO to estimate the effort for each project. Effort would be calculated with the help of the following equation:

$$E= a\ (KLOC)^b \quad (4)$$

In case of the COCOMO organic mode the values of a and b would be 2.4 and 1.05 respectively. We have tabulated the results of FP and efforts after applying the results of [13] and equation-1 in Table II

TABLE II

| Project No. | FP | Effort |
|---|---|---|
| 1. | 218 | 38 |
| 2. | 187 | 32 |
| 3. | 140 | 24 |
| 4. | 93 | 15 |
| 5. | 125 | 21 |
| 6. | 78 | 13 |
| 7. | 156 | 26 |
| 8. | 234 | 41 |
| 9. | 171 | 29 |
| 10. | 203 | 35 |




Graph-01

The relationship between the variable and dependent variable tends to be straight line and the relationship between the FP and the Effort is given in graph-01, which has a highly positive correlation. Therefore a linear regression model $Y= a +bX$ can be assumed to represent the relationship between software effort and function point. From the data set given in Table II, we have found following linear relationship between effort and FP:

$$\text{Effort} = -1.5 + 0.1804\ \text{FP}. \quad (5)$$

For evaluating the different software effort estimation model, the most widely accepted evaluation criteria are the mean magnitude of relative error (MMRE) and the probability of a project having a relative error of less than or equal to 0.25.The magnitude of relative error (MRE) is defined as follows:

$$MRE_i = \text{absolute}\ (\text{Actual Effort}_i - \text{Predicted Effort}_i)/\ \text{Actual Effort}_i \quad (6)$$

The MRE value would be calculated for each observation i whose effort is predicted. The summation of MRE over multiple observations (N) can be achieved through the Mean MRE (MMRE) as follows:

$$MMRE= 1/N \sum_i^N MREi \quad (7)$$

Finally we have tabulated the results of actual effort, predicted effort and MRE in Table III

TABLE III

| **Project No.** | **Actual Effort** | **Predicted Effort** | **$MRE_i$ MMRE= 0.1356** |
|---|---|---|---|
| **1.** | 38 | 37.83 | 0.0192 |
| **2.** | 32 | 32.23 | $7.187 X10^{-3}$ |
| **3.** | 24 | 23.756 | 0.0101 |
| **4.** | 15 | 15.2772 | 0.0184 |
| **5.** | 21 | 21.05 | $2.380 X 10^{-3}$ |
| **6.** | 13 | 12.57 | 0.03307 |
| **7.** | 26 | 26.6 | 0.02307 |
| **8.** | 41 | 40.71 | $7.073 X 10^{-3}$ |
| **9.** | 29 | 29.34 | 0.0117 |
| **10.** | 35 | 35.1212 | $3.462 X 10^{-3}$ |

## VI. Results and Conclusion

In this paper we have predicted the software project effort using linear regression model. On the basis of the data set values we have computed the following regression model.

$$\text{Effort} = -1.5 + 0.1804\ \text{FP}$$

On the basis of the resultant regression model we can compute the following:

1) 0.1804 means that it will cost 0.1804 man-day to finish one function point [14]. When estimating the software effort, firstly we must know the count of function point, and then calculate the project effort according to equation (5). Enterprise can establish their own linear model by using their records. As we know that it is difficult to figure out the count of function point and it will greatly simplify the process of software estimation.
2) In the future we will predict the values of the function point from the effort and in this case we will get the following linear regression model:

$$FP = a + b\text{Effort}.$$

3) In our case study the value of the MMRE is found to be 0.1356.

## REFERENCES


[1] D. Gupta, S. J. Kaushal, and M. Sadiq, "Software Estimation tool Based on Software Engineering Metrics Model," in *Proc. of IEEE International Conference on Management of Innovation of Technology*, Bangkok, Thailand, 2008, pp. 623-628.

[2] G. D. Boetticher, "Using Machine Learning to Predict Project Effort: Empirical Case Studies in Data –starved Domains," *Model based Requirements Workshop*, San Diege, CA, 2001, pp. 17-24.

[3] I. Myrtveit, E. Stensrud, and M. Shepperd, "Reliablity and Validity in Comparative Studies of Softwarte Prediction Models," *IEEE Transaction on Software Engineering*, vol. 31, no. 5, pp. 380-391.

[4] I. Attarzadeh and S. H. Ow, "Proposing a New Software Cost Estimation Model Based on Artificial Neural Network," in *Proc. of the 2nd International Conference on Computer Engineering and Technology (ICCET 2010)*, 2010, pp. 487-491

[5] K. Choudhary. GA Based Optimization of Software Development Effort Estimation. *International Journal of Computer Science and Technology*. [Online]. 1(1) pp. 38-40. Available: http://www.ijcst.Com/wp-content/ themes/ panorama / pdf/kavita.pdf

[6] L. Pickard, B. Kitchenham, and S. linkman, "An Investigation Analysis Techniques for Software Datasets," in *Proc. of Sixth IEEE International Software Metrics*, Symposium, 1999, pp. 1-13.

[7] M. Shepperd and C. Schofield. Estimating Software Project Effort Using Analogies. *IEEE Transaction on Software Engineering*. [Online]. 23(12). pp. 736-743. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00637387

[8] M. Shepperd and C. Empirical. "Predicting with Sparse Data", IEEE Transactions on Software Engineering. [Online]. 27(11). pp. 987-998. Available : http://dl.acm.org/citation.cfm?id=506048

[9] M. Sadiq, and S. Ahmed, "Computation of Function Point of Software on the basis of Average Complexity," in *Proc. of 2nd International Conference on Advanced Computing and Computing Technologies*, Panipat, Haryana, 2007, pp. 591-594.

[10] M. Sadiq and S. Ahmed, "Relationship between Lines of Code and Function Point and its Application in the Computation of Effort and Duration of a Software using Software Equation," in *Proc of International Conference on Emerging Technologies and Applications in Engineering, Technology and Sciences Rajkot*, Gujarat, 2008, India

[11] M. Sadiq, A. Rahman, S. Ahmad, M. Asim, and J. Ahmad, "esrcTool: A Tool to Estimate the Software Risk and Cost," in *Proc. of IEEE Second International Conference on Computer Research and Development*, 2010, pp. 886-890.

[12] M. Sadiq, S. S. Zafar, M. Asim, and R. Suman, "GUI of esrcTool: A Tool to Estimate the Software Risk and Cost," in *Proc. of 2nd IEEE International Conference on Computer and Automation Engineering*, Singapore, 2010, pp. 673-677.

[13] R. S. Pressman, *Software Engineering- A practitioner's Approach*, V. Edition, Mc Graw Hill, pp. 94.

[14] Y. Zheng, B. Wang, Y. Zheng, and L. Shi, "Estimation of Software Project effort based on functional Point," in *Proc. of on 4th International Conference on Computer Science & Education*, pp. 941-943, 2009.


**Mohd. Sadiq** is pursuing Ph.D. in Computer Engineering from National Institute of Technology, Kurukshetra, Haryana. He did Master of Technology in Computer Science and Engineering with specialization in Software Engineering from Aligarh Muslim University (AMU), Aligarh, U.P., India, in 2005 and Bachelor of Engineering in Computer Science and Engineering from Al-Falah School of Engineering and Technology, Dhauj, Faridabad, Haryana, affiliated to Maharishi Dayanand University, Rohtak, Haryana, in 2001with first position in second year. He has more than 6 years of teaching experience and currently he is working as an Assistant Professor of Computer Engineering in Section of Computer Engineering, University Polytechnic, Faculty of Engineering and Technology, Jamia Millia Islamia (A Central University), New Delhi, India. He has published more than 25 research papers in international and national journals like IACSIT International Journal of Engineering and Technology, International Journal of Recent Trends in Engineering, Finland and International Journal of Futuristic Computer Application, Duke University, USA; and in international and national conferences like IEEE international conferences at Singapore, Thailand, Kerala (India), and Bangalore (India). His research interest includes Software Engineering, Data Structure and Algorithms. Mr. Sadiq is a member of International Association of Engineers (IAENG) England, International Association of Computer Science and Information Technology (IACSIT), Singapore and member of Computer Science Teachers Association (CSTA), USA. Mr. Sadiq is a member of the Editorial Review Board of Journal for Computing Teachers (JCT), Buffalo State College, New York, U.S.A., and also the Member of the Editorial Review Board of Journal of Internet and Information System. Victoria Island Lagos, Nigeria. He is the Reviewer of the International Journal of Computer Science and Technology and International Journal of Electronics and Communication Technology. He is also the **Regional Coordinator** of International Journal of Emerging technologies and Applications in Engineering technology and Sciences and International journal of Computer Applications in Engineering, Technology and Sciences. Mr. Sadiq has also chaired the session of IEEE International Conference on Advances in Recent Technologies in Communication and Computing, 2009, Kerala, India. He is also the member of Technical Program Committee of Springer and ACM International Conferences on Advances in Computing and Communication (ACC-2011) , Kochi, Kerala, India.

**Aleem Ali** is working as guest faculty in Computer Engineering Section, University Polytechnic, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi-25; and he is pursuing M.Tech. in Computer Science from Jamia Hamdard, Hamdard University, New Delhi. His research interests are in the areas of Artificial Neural Networks, Data Mining, and Parallel & Distributed Processing. He has taught various papers including Artificial Intelligence, Programming Languages, Data Structures, C/C++/C#, O.S, Compiler. Mr. Aleem Ali has published several research papers in Journal(s) and Conference (s). He is a lifetime member of ISTE. He has more than four years teaching experience.

**Syed Uvaid Ullah** is working with Department of Electrical and Electronics Engineering, Patel Institute of Technology (RGPV), Bhopal, India.

**Shadab khan** was born at Ghaziabad district of U.P. India on 12th December 1980.Currently he is pursuing M.Tech in Computer Science from JSS Academy of Technical Education, Noida affiliated to U.P Technical University. He did Bachelor of Engineering from Al-Falah School of Engineering and Technology in 2004. He has more than 4 years of teaching experience and currently he is working as a Senior Lecturer of Computer Engineering in Department of Computer Engineering at Sunder Deep College of Engineering and Technology (Affiliated to U.P Technical University, Lucknow) Ghaziabad, U.P. India. He has communicated 2 research papers in international conferences. He also attended various Technical Workshops and actively participated in Faculty Development Programs.

**Qamar Alam** is working as Lecturer in Department of Computer Science, Institute of Management Studies, Roorkee. He is MCA from Krishna Institute of Engineering & Technology, Ghaziabad affiliated to Gautam Buddh Technical University, Lucknow and M.Tech from AL-Falah School of Engineering & Technology Dhauj, Faridabad affiliated to Maharishi Dayanand University, Rohtak, (HR). He has an experience of 2 years of Teaching and 1.5 years of industrial experience as a software developer in reputed IT Company. His areas of interest includes Theory of Automate, Soft Computing, Analysis Design and Algorithm and Software Engineering. He may be contacted on 0955703455 and mailed at alamqamar786@yahoo.com