ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 2, April 2014

A Linear Regression Model with Exponential Transformation for Software Effort Estimation

K.P.Manju, Mrs.B.Arthi M.E(Ph.D)
P.G Student , Department of Information Technology , Easwari Engineering College , Ramapuram, Chennai , India.
Assistant Professor, Department of Information Technology , Easwari Engineering College , Ramapuram, Chennai , India

ABSTRACT—Software Engineering plays a vital role in software development process. Software effort Estimation usually takes place in early stage of software life cycle. Estimation relies upon previous experience. Software development based on accurate estimates because badly chosen estimate causes problems during the performance of software development process. A linear Regression model with exponential transformation is proposed to evaluate the estimation of software effort from use case diagrams. This model can be used in early stages of software life cycle to improve the software effort estimation. Linear Regression is used to find out the relationship between variables to get accurate results in software effort. For accurate results in regression, the data obtained in the effort should be normally distributed. For Normal Distribution exponential transformation of data is applied. The exponential function is used to model the relationship in which a constant change in the independent variable gives the same proportional change in the dependent variable. Linear Regression is applied on normalized data to improve the accuracy of software effort estimation.

INDEX TERMS—Software effort estimation, Use case points Exponential transformation, Linear Regression.

I. INTRODUCTION

Software estimation is an important process that happens during the software project lifecycle. Estimation of Software size, Resources and Cost are included in this estimation process. Software effort estimation is used to estimate the amount of effort required to develop a software projects. Estimation of software effort in early stages of software life cycle is challenging and complex in software companies. Most of software project failures occur due to improper software size, uncertainty of system and software requirements during requirement stage. There are many software effort estimation models which are used to predict the effort in the initial stage of software life cycle. According to Boehm[8], software effort estimation methods are classified into six categories: parametric models, expert judgment, learning oriented techniques which include the case-based reasoning (CBR) method, regression based methods; dynamics based models, and composite methods. Improper accurate effort estimation leads to poor budgeting and planning.

To improve the accuracy of effort estimation, proper communication is needed that helps the managers to allocate the resources efficiently which is done in the initial stages of project development. Estimating the size of the software project includes use case points, function points, object points .Accurate estimation of software size is complex in software development for determining the quality, efficiency of the software projects.

In this paper we focus on better accurate estimation results based use case diagrams are given as input for software size and use case points is taken as output. A linear regression model with exponential transformation is used to find out the relationship between the variables that includes software size and effort for improving the accuracy in software effort estimation.

The next section explains related research for our project,In section 3 describes the proposed methodology. Section 5 describes results and conclusion

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 2, April 2014

II. RELATED RESEARCH

During the initial stage of project development, the effort estimation is essential for the software industry. To meet the demands of the industry, software estimation should be accurate and reliable.

Nassif A.B and et. al[2] proposed a regression model to estimate the software effort based on use case diagrams as input and produce use case points as output for software projects. Regression analysis is applied to find out the relationship between software effort and size with logarithmic transformation to improve the accuracy of software effort estimation.

Wang F and et.al [3] proposed a extended use case point method for software effort estimation to solve the uncertainty and abrupt classification of use cases. Extended Use case point method proposed an approach by integrating with fuzzy logic and Bayesian Belief networks which provides probability distribution and classification of gradual refinesin software effort estimation.

Lopez-Martin and et.al [4] proposed a general regression neural network for prediction of software effort from industrial projects with dataset obtained in ISBSG repository. To improve the accuracy of software effort evaluation is done in magnitude of relative error (MMRE) and Mean Absolute Error(MAE).Many projects have been trained to validate the neural network model. Inputs to the neural network model depend upon the use case points, function points, Lines of code to train the data.

Pendharkar P.C and et.al [5] proposed a probabilistic model known as Bayesian model which provides point forecast for predicting software development effort. It uses previous dataset projects to improve the software effort. The proposed model used in finding the relationships and missing values for forecasting the software effort. For estimating risk factors new tools have been developed to solve the decision making risk factors by Bayesian model which improves the forecasting of software effort.

Jiang Z and et.al [6] proposed the effects of software size on software development effort and software quality. Function points are the inputs to the software size which are used in prediction of software development effort. It depends upon the high software quality which leads to delivery of project within time and maintenance of costs. Estimating the effort required in each stage of development does not leads to projects failures and increases the software quality. Various data points have been evaluated which is used in the development of software effort.

III. PROPOSED METHODOLOGY

In this section a proposed regression model is described.As shown in fig .1. dataset is prepared by taking the use case diagrams from software application as input.The use case diagram represents the functional requirements of the system.Use case points is calculated by identifying the number of actors and number of usecases.Effort estimation is calculated by multiplying each use case point size with 20 persons hours proposed by karner [1].

After computing the effort ,the data obtained must be normal distributed.If the data is not normalized,normalize the data using exponential transformation.After exponential transformation,linear regression is applied to find out the relationship between the exp(size) and exp (effort) which improves the accuracy of software effort estimation.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 2, April 2014

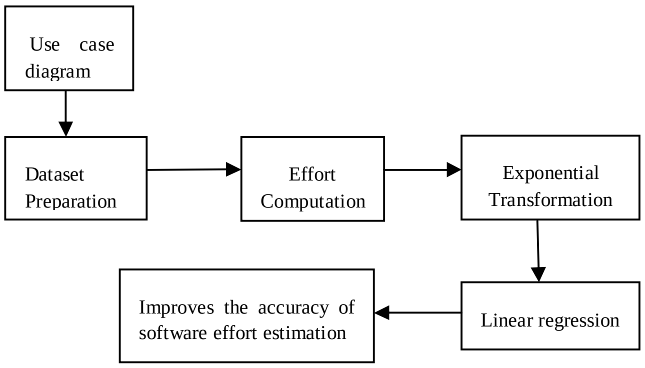
Use case diagram
Dataset Preparation
Effort Computation
Exponential Transformation
Linear regression
Improves the accuracy of software effort estimation

Fig 1. System Architecture

A. Use case points

Use case point model is one of the methods to estimate the software size at the early stages of software life cycle. Use Case point model consists of two stages unadjusted use case points (UUCP), adjusted use case points (UCP). Unadjusted points are calculated by adding the two terms unadjusted actor weight and unadjusted use case weight.

(a) Unadjusted actor weight (UAW):

UAW is calculated by counting the number of actors involved in the use case diagrams. As shown in table 1 actors are classified as simple, average, complex and multiplied by corresponding complexity weights.

TABLE 1. COMPLEXITY WEIGHTS OF ACTORS[1]

Actor Complexity	Description	Weight
Simple	Actorcommunicates with another system using a defined application interface	1
Average	Actor communicates with another system interacting with protocol(TCP/IP)	2
Complex	Actor interacting using graphical user interface	3

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 2, April 2014

UAW is represented as shown in the equation (1)

UAW= (Number of simple actors*1) + (Number of average actors*2) + (Number of complex actors*3) (1)

TABLE 2.COMPLEXITY WEIGHTS OF USE CASES [1]

Use case Complexity	No of transactions	Weight
Simple	Less or equal to 3	5
Average	Between 4 to 7	10
Complex	More than 7	15

(b)Unadjusted Use case Weight (UUCW):

UUCW is calculated by counting the number of use cases in the use case diagram. As shown in table 2 use cases are classified as simple, average, complex based upon the number of transactions. Complexity weight is assigned based upon the use case transactions. UUCW is represented as shown in the equation (2)

UUCW = (Number of simple use cases *5) + (Number of average Use cases *10)+ (Number of complex Use cases *15) (2)

(c) Unadjusted use case points (UUCP):

Unadjusted Use case points are calculated by adding the unadjusted actor weight and unadjusted use case weight .UUCP is shown in the equation(3)

$$UUCP = UAW+UUCW \quad (3)$$

(d) Adjusted Use case points (UCP):

After calculating the unadjusted use case points, adjusted use case points is calculated by multiplying with environmental factors and technical factors.

TABLE 3.ENVIRONMENTAL FACTORS [1]

Factor	Description	Weight
E1	Part-Time Workers	-1
E2	Application Experience	0.5
E3	Object oriented Experience by the team	1
E4	Analyst Capability	0.5
E5	Motivation by the team	1
E6	Use of stable requirements	2
E7	Familiar with the model used in development process	1.5
E8	Difficulty in programming Language	-1

TABLE 4.TECHNICAL FACTORS [1]

Factor	Description	Weight
T1	Distributed system	2
T2	Performance	2
T3	End User efficiency	1
T4	Internal Complex complexity	1
T5	Reusability	1
T6	Easy installation	0.5
T7	Usability	0.5
T8	Portability	2
T9	Changeability	1
T10	Concurrent use	1
T11	Security	1
T12	Access to third parties	1
T13	Training facilities required	1

(e) Environmental Complexity Factor (ECF):

Environmental complexity factor (ECF) is used to adjust the size based on environmental considerations of the system. As shown in table 3 it depends upon 8 environmental factors (E1-E8) and weight is assigned to each factor based upon its efficiency. The environmental factor is shown in the equation (4)

$$ECF = 1.4 + (-0.03 * \text{Environmental Factor}) \quad (4)$$

(f) Technical Complexity factor (TCF):

Technical complexity factor (TCF) is used to adjust the size based on technical considerations of the system. As shown in table 4 It depends upon 13 technical factors (T1-T13) and weight is assigned to each factor based upon its complexity. The technical factor is shown in equation (5).

$$TCF = 0.6 + (0.01 * \text{Technical factor}) \quad (5)$$

(g) Use case points:

Thus the adjusted use case points (UCP) is calculated from the unadjusted use case points (UUCW and UAW), technical factor (TCF) and environmental factor (ECF) has been determined. Use case points (UCP) is calculated as shown in the equation (6)

$$UCP = (UUCW + UAW) \text{ x } TCF \text{ x } ECF \quad (6)$$

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 2, April 2014

(h) Software effort:

Based on Karner, software effort is calculated as shown in equation (7)

$$\text{Effort} = \text{Size}*20 \tag{7}$$

Size is calculated from UCP and effort is measured in person hours.

B.Regression model:

Software effort depends upon software size, project complexity team productivity. Software effort is represented as follows [7].The equation is represented as shown in equation (8)

$$\frac{\text{Complexity}}{\text{productivity}}*\text{size} \tag{8}$$

The proposed regression model used to find out the relationship between software size and effort. To improve the accuracy of software effort estimation data should be normally distributed. Regression analysis is applied on the normalized data. If data is not normalized exponential transformation is applied between size and effort as shown in the equation (9)

$$\exp(\text{effort}) = \exp(\text{size})+c \tag{9}$$

The regression model depends on the software size increases, software effort increases. The software effort equation of regression model with exponential transformation is shown in the equation (10)

$$\text{Effort} = 8.16 \text{ x}\frac{\text{complexity}}{\text{productivity}}\text{x (size)}^{1.17} \tag{10}$$

Effort in person hours, size is use case points, Project complexity, productivity depends upon environmental factors.

IV RESULTS AND DISCUSSION

In this section, we discuss the results of our work.To evaluate the performance of the proposed model linear regression model with exponential transformation and log linear regression models. To predict the software effort from use cases a comparison was conducted between linear regression model with exponential transformation and log linear regression model. It shows the relationship between the software size and effort as shown in fig 2. To improve the accuracy of software effort estimation the evaluation process is based upon Mean of Magnitude of Error Relative (MMER), Predication Level (PRED), Root Mean Squared Error (RMSE), Mean absolute error (MAE),Standard deviation(SD).Fig 3 shows the accuracy is obtained by calculating the difference between actual effort and reg-effort.

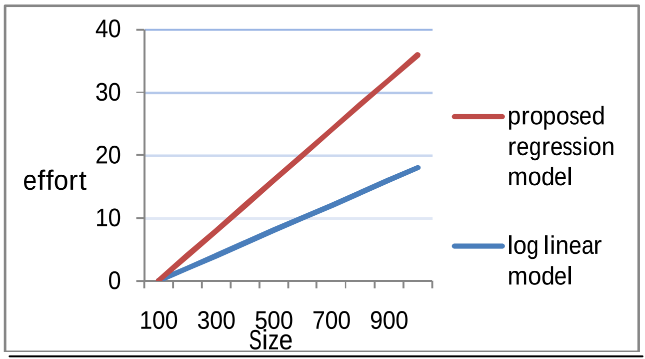

Fig 2. RELATIONSHIP BETWEEN SOFTWARE SIZE AND EFFORT

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 2, April 2014

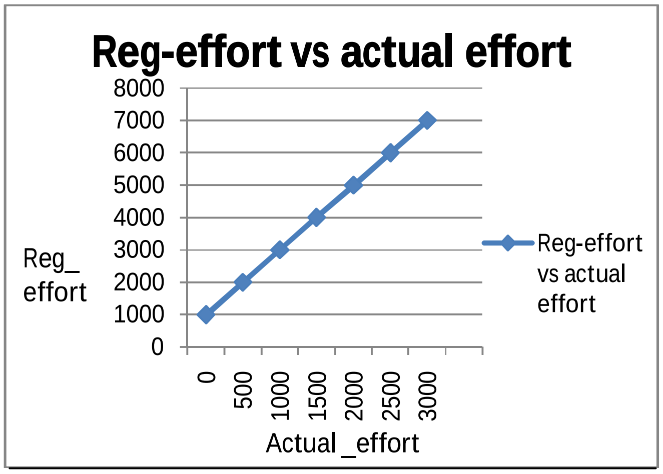

Fig 3. REG_EFFORT VS ACTUAL EFFOR

V CONCLUSION

The use case point model is used in the early stage of software life cycle to conduct early software effort estimations. The main advantage of this model is that it is simple and can be easily automated. In this paper use case points are given as input for software size and we have analyzed that log linear regression model and linear regression model with exponential transformation used for software effort estimation. The proposed regression model in this paper used to tackle the disadvantages and used to improve the accuracy of software effort estimation.

REFERENCES

[1] Karner,G, .Resource estimation for objectoryprojects. Objective Systems.1993
[2] Nassif, A.B., Ho, D., Capretz, L.F., . Regression model for software effort estimation based on the use case point method. In: 2011 International Conference on Computer and Software Modeling, Singapore, pp. 117–121.2011a
[3] Wang, F., Yang, X., Zhu, X., Chen, L., Extended use case points method for software cost estimation. In: International Conference on Computational Intelligence and Software Engineering2009.
[4] Lopez-Martín, C., Isaza, C., Chavoya, A.,. Software development effort prediction of industrial projects applying a general regression neural network. Empirical Software Engineering 17,pp 1–19.2011
[5] Pendharkar, P.C., Subramanian, G.H., Rodger, J.A., A probabilistic model for predicting software development effort. IEEE Transactions on Software Engineering 31,pp 615–624.2005.
[6] Jiang, Z., Naudé, P., Jiang, B.,. The effects of software size on development effort and software quality.International Journal of Computer and Information Science and Engineering 1, 230–234.2007
[7] D. D. Galorath and M. W. Evans, Software Sizing, Estimation, andRisk Management. Boston, MA, USA: Auerbach Publications, 2006
[8] Boehm, B.W.,Software Engineering Economics. Prentice-Hall.1981