Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Prediction of Software Development Cost and Effort using Multiple Linear Regression

R. Venkatesh Kumar
Research Scholar
Department of Statistics
Madras Christian College
Tambaram, India

R. Chandrasekaran
Associate Professor and Head (Retd.)
Department of Statistics
Madras Christian College
Tambaram, India

Abstract - Estimation of the software development cost and effort are of importance in the software industry. These two factors mainly lead to the success or failure of a project. Many industry experts have worked to develop useful models to predict the accurate cost and effort required for a software product development for 40 years now, but still there is not enough evidence for an efficient effort estimation method. Hence we tried to develop an estimation model for better software project management. The main objective of this research is to identify the groups of software development projects with significant comparable characteristics based on the parameters ADDED, MODIFIED, REUSED and DELETED of Source Lines of Code (SLOC) which are in turn associated with the ACTUALEFFORT. The software development projects are clustered by using hierarchical clustering techniques. Our new models have been developed based on the clustered groups to predict software effort from source lines of code. The method used to build these models is Multiple Linear Regression (MLR). The unsupervised classification method used in this paper would identify the similar category of projects and forecast the software development cost and time effort. Hence, this approach will be useful for planning and preventive actions in the management of software development projects.

Keywords: Effort Estimation, Cluster Analysis, Average Linkage Method, Median Method, Ward Method, Source Lines of Code, Multiple Linear Regression.

I. INTRODUCTION

The software industry experts intend to develop useful models to predict accurate cost and effort for a software product development for over 40 years now. A number of models have been fabricated for the software development cost and effort estimation since then (Suri and Pallavi, 2012)but till date no precise model has been developed to provide enough evidence for an efficient software development effort estimation method in spite of investing considerable amount of money, time and activities.

Almost 44% of the software projects were delivered late and are over budgeted according to Standish Group. This is an indicator for increasingly important role of project management (Kotonya and Sommerville, 1998; Demirors and Gencel, 2004). The International Society of Parametric Analysis (ISPA) recognized the main reasons behind the software project failures (Eck and et.al, 2008) and these can be summarized as follows:

- Shortness in understanding the requirements
- Inappropriate software size estimation
- Lack of estimation of the resource's skill level

In 2009, Standish Group conducted another study related to software estimation. According to the study, some other factors also impact the software project failures and the factors includes;

- Unrealistic estimation
- Ignoring historical data
- Uncertainty of system and software requirements
- Unskilled estimators
- Budget limitation
- Optimism in software estimation

The fact that many software projects fail due to inaccuracy of estimation forced the researchers to conduct investigation on software development effort estimation to arrive at a better software effort and cost assessment. In general the overall software project cost and effort are decided based on the development effort.

This paper is organized as follows: Section II covers the literature review about the software estimation methods. Section III provides a detailed explanation about unsupervised classification method and Multiple Linear Regression (MLR). Section IV gives the novel model developed based on a real time software development data, and finally, Section V presents conclusion of the present research.

II. LITERATURE REVIEW

In the area of software cost/effort estimation, considerable number of studies is reported and hence there are several methods available for the estimation and these methods can be categorized as Algorithmic and Non-algorithmic (Musilek, et al. 2002; Yahya, et al. 2008; Lavazza and Garavaglia 2009; Yinhuan et al. 2009; Sikka et al. 2010; Khatibi and Jawawi, 2011; Chandrasekaran and Venkatesh, 2012). Each of these methods has its own pros and cons. For an efficient estimate, the requirements should be known better. There are several parameters which affect the software estimation like size of the software product applicability, category to which the product serve, domain of applicability or development etc. and in general these parameters need to be considered while selecting the method of estimation.

Shepperd and Schofield (1997), using analogy proposed a software estimation model which was evaluated based on 275 projects from nine different industrial datasets. The author's stress that estimation models based upon analogy outperform any other algorithmic models based on stepwise regression.

Jorgensen et al. (2003) applied regression toward the mean (RTM) method with analogy for software effort estimation. The proposed model was evaluated based on 5 different datasets. The authors insisted that the accuracy of software effort estimation using analogy would be improved when using RTM.

Jiang et al. (2007) and Xia et al. (2008) using ISBSG data, had built linear regression models with a logarithmic transformation based on function points. Regression model had been used as an activation function in a neural network by Xia et al. for the calibration of weights in the function point model. However, regression model was used to study the effect of software size on development effort and software quality by Jiang et al. But these models ignore the influence of the non-functional requirements on estimation and this is the main drawback of these models.

Tan et al. (2009) suggested a novel LOC estimation method for information systems based on the theoretical data models through a multiple linear regression model. The authors evaluated their work using open source and industrial projects.

Nassif et at. (2012) proposed new regression model based on the use-case-point-size metric which use case diagrams as input and results in the software size in use-case-points as output. The effort equation proposed also considers the non-linear relationship between software size and effort, as well as project's complexity and productivity influence. According to this study the software effort estimation accuracy can be improved by 16.5% using PRED(25) and 25% using PRED(35).

Kusuma et al. (2014) provides an overview of existing software cost estimation models and their techniques. According to this study, none of the method is necessarily better or worse than the other method, in fact, their strengths and weaknesses are often complimentary to each other. To understand their strengths and weaknesses is critical while estimating the software projects.

III. MULTIPLE LINEAR REGRESSIONS

The Multiple Linear Regression (MLR) is a statistical technique that uses several explanatory variables (Independent Variables) to predict the outcome of a response variable (Dependent Variable) and also it is an extension of a simple linear regression model. Due to its power and flexibility, multiple regression modeling has become an important technique of statistical analysis in many fields. The foremost objective of MLR is to model the relationship between the explanatory and response variables (Anderson, et al., 2009). The MLR is an experimental model and belongs to algorithmic group of techniques. Thus to estimate the present project, this model requires data from the past projects (Boehm, et al., 2000; Singh, et al., 2008).

MLR model is defined as:

$$\mathbf{Y} = \boldsymbol{\alpha} + \boldsymbol{\beta_1 X_1} + \boldsymbol{\beta_2 X_2} + \ldots + \boldsymbol{\beta_p X_p} + \boldsymbol{\varepsilon} \qquad \textbf{EQ. (1)}$$

where,

$\boldsymbol{\alpha}$, is called as constant term; $\boldsymbol{\beta_1}$, $\boldsymbol{\beta_2}, \ldots, \boldsymbol{\beta_p}$ are the coefficients relating to the $\mathbf{p}$ predictors variables and $\boldsymbol{\varepsilon}$, is called as the residual or error.

IV. METHODOLOGY

This section introduces a novel model to predict software effort estimation from source lines of code. As a first step in this research Hierarchical clustering techniques namely, **Average Linkage Method**, **Median Method** and **Ward Method** are used to find the similar characteristic of software effort from the source lines of code (ADDED, MODIFIED, REUSED and DELETED). Our new model is developed based on the clustered groups to predict software effort estimation. The method used to build these models is Multiple Linear Regression (MLR).After data validation and normalization, there are 937 historical development projects considered as sample data, which are used for the present analysis. The Statistical Software Package IBM SPSS 19.0 is used for the present research. The number of projects in each technology is displayed in the *Figure 1*.

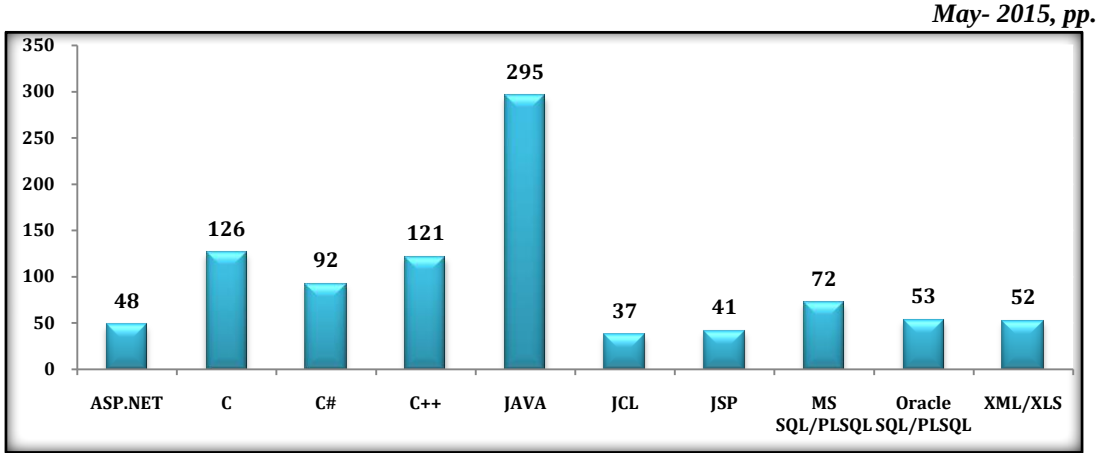

Fig 1. Sample Sizes of Technologies

For the analysis of Cost and Effort Estimation Model, the variables in *Table 1* have been carefully selected by carrying out various rigorous analyses; review of previous works and also by referring various International articles and journals.

Table.1 Variables included in the study

Variables	Description	Variable Utilization
ADDED	Source Lines of Code Added	Independent Variable
MODIFIED	Source Lines of Code Modified	Independent Variable
REUSED	Source Lines of Code Reused	Independent Variable
DELETED	Source Lines of Code Deleted	Independent Variable
EFFORT	Actual Effort spent (hrs)	Dependent Variable

A. Classification of Software Projects

Cluster analysis, also called as unsupervised classification, is a technique by which a set of objects are grouped in such a way that objects with similar characteristics belong to the same group (called a cluster). To cluster data, various procedures are available. Among them most common methods are hierarchical cluster analysis, *k*-means cluster, and two-step cluster. In our study, we have used *Average Linkage*, *Median Method* and *Ward Method* hierarchical clustering methods. In the software industry, these methods are the ones which are widely used to classify the projects with similar characteristics. To group these projects, actual effort (EFFORT) and size (ADDED, MODIFIED, REUSED and DELETED) were used as the input to the hierarchical clustering methods. In the final step, a single cluster has been formed by joining the two groups, formed in the last but one step (*Figure 2*).The first group includes the languages/technologies *MS SQL/PLSQL, Oracle SQL/PLSQL, ASP.NET, JSP, XML/XSL* and *JCL* which are often used for web/database development & management and hence these are referred as *Web/Database development technologies.* The second group contains the languages/technologies *C#, Java, C* and *C++*, which are mostly used for Application development and these can be termed as *Application Development Technologies*.

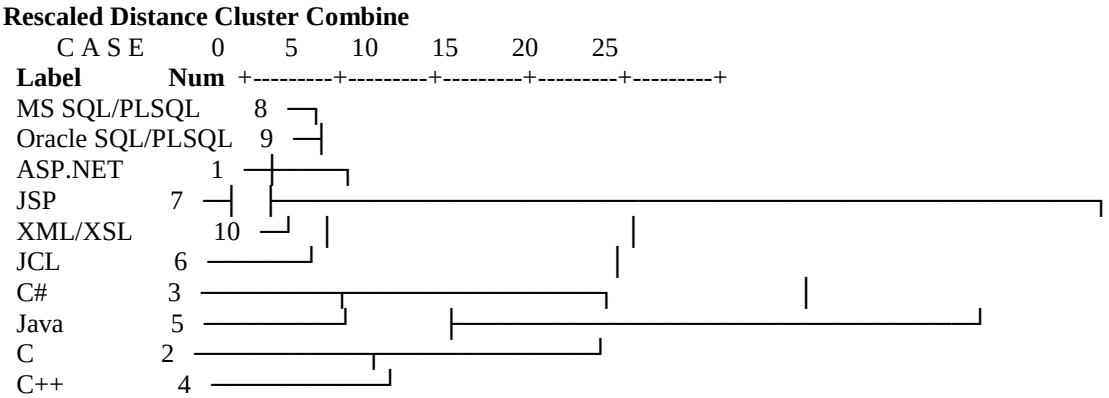

Fig 2. Dendrogram

All the three clustering methods (*Average Linkage, Median Method* and *Ward Method*) arrived at the same result, and hence the cluster formed can be considered as the natural cluster.

B. Estimation Models

In this section, a novel model to foretell the effort estimation of two groups, Application development projects and Web/Data based development projects based on Multiple Linear regression model (MLR) is proposed. The variables

which are used to predict the effort estimation are detailed in *Table 1*. The independent variables to be included in the multiple linear regression (MLR) models can be obtained by several procedures. One method is to include all the relevant variables. Another method is to use stepwise procedure – backward regression, forward regression and stepwise regression (Kvanli and et al., 2006). The present study uses the stepwise model, which is more popular than the other two. This stepwise method begins by including one variable (Xi) that has the highest correlation with Y and keeps on including the independent variables one at a time.

C. Effort Estimation for Application Development

Based upon the clusters formed, 634 software development projects clustered together is considered for effort estimation of that group, namely application development projects, since most of the technologies included in the cluster is widely used for application development. The four software techniques namely, *C#, Java, C and C++* are combined in this group. The *Table 2* Model summary contains the Application development projects using the stepwise Regression procedure. The first column "**Model**" refers to the stage in the hierarchy, where independent variables Size (ADDED, MODIFIED, REUSED and DELETED) has been used as a predictor and in this model all independent variables are used to forecast the effort. The Second column of this table "**R**" refers to the correlation between Effort and Size variables. The third column in this table "$\mathbf{R^2}$" (or coefficient of determination) is used to evaluate the adequacy of the model (Montgomery, et al, 2006) and acceptable value of $\mathbf{R^2}$ is ≥ 0.5 (Humphrey, 1995).The $\mathbf{R^2}$ reported for this regression model is 0.667, approximately 67 % of the variation in Effort can be explained by the variables ADDED, MODIFIED, REUSED and DELETED. This shows that the relation between Effort and Size. The fourth column, "**Adjusted $\mathbf{R^2}$**" is intended to "control for" overestimates of the coefficient of determination. The Fifth column of this table, "**Std. Error of the Estimate**" is the standard deviation of the residuals.

Table.2 Model Summary for Application Development

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Predictor Variables
1	.742	0.551	0.55	0.108	ADDED
2	.788	0.621	0.62	0.099	ADDED, MODIFIED
3	.813	0.661	0.659	0.094	ADDED, MODIFIED, REUSED
4	.817	0.667	0.665	0.093	ADDED, MODIFIED, REUSED, DELETED

Dependent variable : Effort

Table 3 depicts the ANOVA for the regression models. Examination of the last two columns of the output shown in the ANOVA table, indicates that the models are good enough, since p-value $< 0.001 \leq 0.05$. At the $\propto = 0.05$ level of significance, there exists enough evidence to conclude that the predictors are useful for predicting the Actual Effort. The number of variables that were added during the final step is indicated by examination of the *df* column. (i.e., the degree of freedom counts of the number of predictors in the model).

Table.3 ANOVA for Application Development

Model		Sum of Squares	DF	Mean Square	F	Sig.	Predictor variables
1	Regression	9.076	1	9.076	775.393	.000[a]	ADDED
	Residual	7.398	632	0.012			
	Total	16.474	633				
2	Regression	10.238	2	5.119	517.919	.000[b]	ADDED MODIFIED
	Residual	6.236	631	0.01			
	Total	16.474	633				
3	Regression	10.884	3	3.628	408.87	.000[c]	ADDED MODIFIED REUSED
	Residual	5.59	630	0.009			
	Total	16.474	633				
4	Regression	10.986	4	2.746	314.757	.000[d]	ADDED MODIFIED REUSED DELETED
	Residual	5.488	629	0.009			
	Total	16.474	633				

*Table 4*displays the coefficients results. The column un-standardized coefficient provides the estimated values of the regression coefficients 'B' and their standard errors. To evaluate a variable's reliability, "***Collinearity Statistics***" is also important. Collinearity (also called asmulti-collinearity) specifies the assumption that the independent variables are uncorrelated (Darlington, 1968).

The Tolerance level for correlation ranges is from 0 to 1, i.e., 0 (no independence) to 1 (completely independent).The Variance Influence Factor (VIF) is an index of the amount of variance of each regression coefficient, getting bigger with uncorrelated independent variables (Keith, 2006). The VIF associated is large, when a predictor variable has a strong linear association with other predictor variables, and gives evidence of multicollinearity. The rule of thumb for a large VIF value is greater than10 (Keith, 2006; Shieh, 2010). *Small values for tolerance and huge VIF values show the presence of multicollinearity*. There is no collinearity in this model which is proved by the coefficients table and this model is following by MLR assumptions.

Table.4 Coefficients for Application Development

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Collinearity Statistics	
		B	Std. Error	Beta			Tolerance	VIF
1	(Constant)	0.055	0.005		10.219	0.000		
	ADDED	0.816	0.029	0.742	27.846	0.000	1.000	1.000
2	(Constant)	0.046	0.005		9.033	0.000		
	ADDED	0.752	0.028	0.685	27.316	0.000	0.955	1.047
	MODIFIED	0.437	0.040	0.272	10.840	0.000	0.955	1.047
3	(Constant)	0.043	0.005		8.897	0.000		
	ADDED	0.724	0.026	0.659	27.530	0.000	0.940	1.064
	MODIFIED	0.404	0.038	0.251	10.535	0.000	0.946	1.058
	REUSED	0.417	0.049	0.201	8.535	0.000	0.968	1.033
4	(Constant)	0.041	0.005		8.678	0.000		
	ADDED	0.717	0.026	0.653	27.412	0.000	0.934	1.070
	MODIFIED	0.368	0.039	0.229	9.335	0.000	0.879	1.138
	REUSED	0.413	0.048	0.199	8.521	0.000	0.968	1.033
	DELETED	0.198	0.058	0.082	3.415	0.001	0.910	1.099

a. Dependent Variable: EFFORT

Figure 3 displays the histogram of the residuals with a normal curve and the residuals look quite close to normal. The Normal probability plot is displayed in the *Figure 4.*

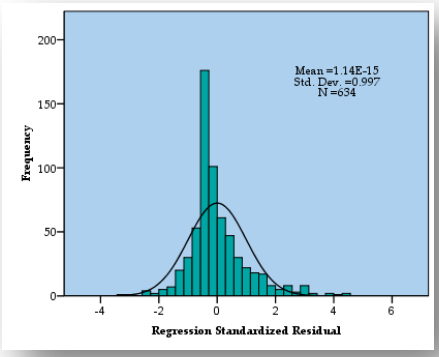

Fig 3. Histogram of the residuals

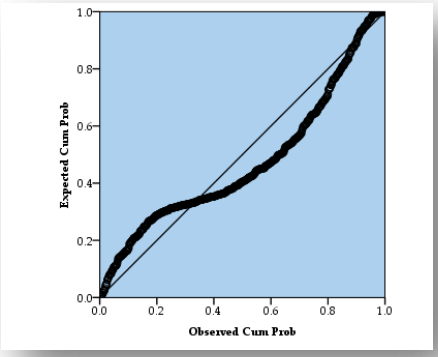

Fig 4.Normal P-P Plot of Regression Standardized Residual

Once the assumption of Normality is satisfied, the residuals form a random sample from a standard Normal distribution, and the plot will be a straight line from the origin (0,0) to top right (1,1). This residual is quite a good approximate as it is close to the straight line, which means we can rely on the regression coefficients resulted.

From the above results, it is deduced that the proposed multiple linear regression equation is valid and it is used to determine the effort by using the Size (ADDED, MODIFIED, REUSED and DELETED) of application development projects and the model Equation (2) is given by,

Effort = 0.041 + 0.717 (ADDED) + 0.368 (MODIFIED) + 0.413 (REUSED) + 0.198 (DELETED) **EQ. (2)**

In the final step of the analysis, the independent variables are entered into the regression equation. Based on the coefficients result, the independent variables are significantly related to Actual Effort, F (4, 629) = 314.757, p <.001. The multiple correlation coefficient is 0.817 and the $\mathbf{R^2}$ (coefficient of determination) is 0.667, which means that approximately 67% of the variance of the Actual Effort could be accounted by **ADDED, MODIFIED, DELETED** and **REUSED**.

D. Effort Estimation for Web/Database Development

The first cluster formed contains 303 software development projects and is named as Web & database development projects since most of the technologies included in the cluster are widely used for Web or database development. There are six software techniques *MS SQL/PLSQL, Oracle SQL/PLSQL, ASP.NET, JSP, XML/XSL* and *JCL* is included in this group. In *Table 5,* Variables Entered/Removed shows that the independent variables ADDED, MODIFIED and DELETED have been used as predictors and in this model RESUSED variable has not been included as the "p" is 0.927 i.e. this variable is statistically not significant for this model. The independent variables ADDED, MODIFIED and DELETED are sufficient to predict the effort for Web & database development projects.

Table. 5 Variables Entered/Removed

Model	Variables Entered	Method
1	ADDED	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	MODIFIED	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	DELETED	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

Dependent Variable: EFFORT

The *Table 6* Model summary depicts the Web & database development projects' using the stepwise Regression procedure. The first column "**Model**" refers to the stage in the hierarchy of the independent variables. The third column of this table "$\mathbf{R^2}$" (or coefficient of determination) reported for this regression is 0.562 and hence approximately 56% of the variation in Effort can be explained by the variables ADDED, MODIFIED and DELETED. This shows a relation between Effort and size. As mentioned earlier, the larger the R^2, the smaller this will be relative to the standard deviation of the criterion.

Table.6 Model Summary for Web/Database Development

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Predictor Variables
1	.676	0.457	0.455	0.072	ADDED
2	.729	0.531	0.528	0.067	ADDED, MODIFIED
3	.749	0.562	0.557	0.065	ADDED, MODIFIED, DELETED

Dependent Variable: EFFORT

*Table 7*presents the ANOVA for the regression models. From the "p" or sig-value of ANOVA, we notice that there is a significant relationship among the variables at the 99% confidence level. At the $\propto = 0.05$ level of significance, there exists enough evidence to conclude that the predictors are useful for predicting the Actual Effort.

Table.7 ANOVA for Database Development

Model		Sum of Squares	df	Mean Square	F	Sig.	Predictor Variables
1	Regression	1.330	1	1.330	253.406	.000[a]	ADDED
	Residual	1.579	301	0.005			
	Total	2.909	302				
2	Regression	1.546	2	0.773	170.091	.000[b]	ADDED MODIFIED
	Residual	1.363	300	0.005			
	Total	2.909	302				

	Regression	1.633	3	0.544	127.64	.000[c]	ADDED
3	Residual	1.275	299	0.004			MODIFIED
	Total	2.909	302				DELETED

Dependent Variable: EFFORT

Table 8 presents the regression coefficients. Looking upon the VIF and tolerance, there is no collinearity in this model and this model follows MLR assumptions.

Table.8 Coefficients for Web/Databases Development

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Collinearity Statistics	
		B	Std. Error	Beta			Tolerance	VIF
1	(Constant)	0.013	0.005		2.632	0.009		
	ADDED	0.531	0.033	0.676	15.919	0.000	1.000	1.000
2	(Constant)	0.007	0.005		1.635	0.103		
	ADDED	0.498	0.031	0.634	15.861	0.000	0.977	1.024
	MODIFIED	0.255	0.037	0.276	6.897	0.000	0.977	1.024
3	(Constant)	0.009	0.004		1.932	0.054		
	ADDED	0.443	0.033	0.563	13.487	0.000	0.840	1.190
	MODIFIED	0.236	0.036	0.255	6.527	0.000	0.963	1.038
	DELETED	0.226	0.05	0.190	4.534	0.000	0.836	1.196

a. Dependent Variable: EFFORT

Figure 5 displays the histogram of the residuals with a normal curve and *Figure 6* displays the Normal probability plot.

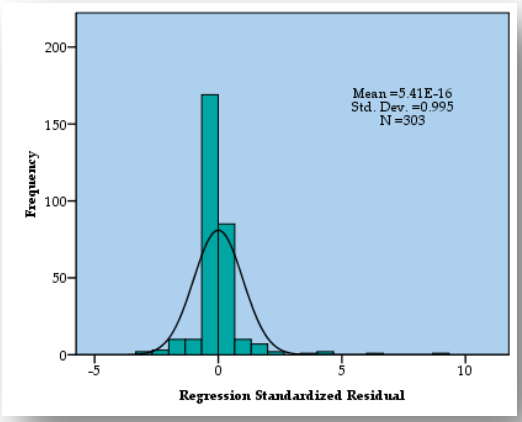

Fig 5. Histogram of the residuals

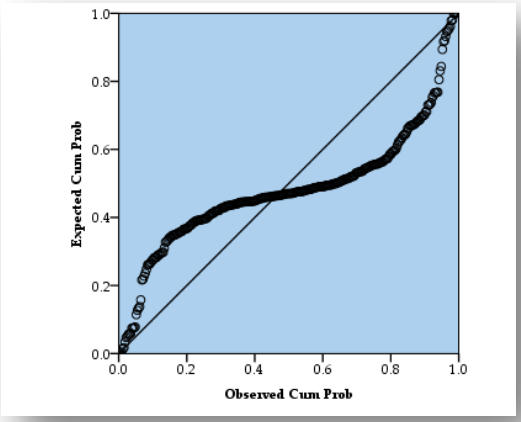

Fig 6.Normal P-P Plot of Regression Standardized Residual

From the above results, it is deduced that the proposed multiple linear regression equation is valid and it can be used to determine the Effort using Size (ADDED, MODIFIED and DELETED) for Web & Database development projects.

The final regression equation is given as.

Effort = 0.009 + 0.443 (ADDED) + 0.236 (MODIFIED) + 0.226 (DELETED) **EQ.(3)**

Based on the coefficients result, the independent variables are significantly related to Actual Effort, F (3, 299) = 127.640, p <.001. The multiple correlation coefficient is 0.749 and the $\mathbf{R^2}$ (coefficient of determination) 0.562, means that approximately 56 % of the variance of the Actual Effort could be accounted by **ADDED, DELETED** and **MODIFIED.**

V. CONCLUSION

In the present world, software industry is more complex and more matured but still in constant search for a reliable tool to measure the processes so that the cost can be reduced and the profit can be maximized. Forecast for accurate effort and the required schedule for any development project are still a challenge in the software industry. Generally, market demands initiate a Software product to be developed. Marketing and Salespeople bring in these kinds of client

requirements to a software company which necessitates a R&D of the product requirements. Based on the research done and the client requirements, Senior Business Analyst creates architecture along with functional and technical design specification document for the product. Then the software development plan and effort/schedule estimation will be created by the project managers. Success of any software development product depends on project planning and hence it is crucial; poor or lack of planning will leads to an unfavorable outcomes and hence, the failure of the project. Our main aim is to predict the effort estimation based on the lines of code so that the planning and scheduling can be done effectively and efficiently and this is achieved well by developing new models and a new software application. We have used multiple regression models and fitted the regression equations for Application development and web/database development projects. Anew estimation software application is also developed, based on these models which will help to the project management to estimate the effort required before initiation of the actual development. These will equip the project managers to effectively plan and schedule the project.

REFERENCE

[1] Anderson, D. R., Sweeney, D. J., and Williams, T. A., 2009, "Statistics for Business and Economics". Thomson South-Western.

[2] Boehm, B.W., Abts, C., Clark, B., and Devnani-Chulani, S., 2000, "COCOMO II Model Definition Manual". Version 2.1, The University of Southern California.

[3] Chandrasekaran, R., and Venkatesh kumar, R., 2012, "On the Estimation of the Software Effort and Schedule using Constructive Cost Model – II and Functional Point Analysis", *International Journal of Computer Applications*, Vol. 44, No.9, pp. 38-44.

[4] Darlington, R. 1968, "Multiple regression in psychological research and practice", *Psychological Bulletin*, Vol. 69, No.3,pp.161-182.

[5] Demirors, O., and Gencel, C., 2004, "A Comparison of Size Estimation Techniques Applied Early in the Life Cycle", *Software Process Improvement*, Vol. 3281, pp. 184-194.

[6] Eck, D., Brundick, B., Fettig, T., Dechoretz, J., and Ugljesa, J., 2008, "Parametric estimating handbook", *The International Society of Parametric Analysis*, Fourth Edition.

[7] Humphrey, W., 1995, *A Discipline for Software Engineering.* Addison Wesley.

[8] Jiang, Z., Naude, P., and Jiang, B., 2007, "The effects of software size on development effort and software quality", *World Academy of Science, Engineering and Technology*, Vol.34, pp. 31-35.

[9] Jørgensen, M., Indahl, U., and Sjøberg, D., 2003, "Software effort estimation by analogy and regression toward the mean" *Journal of Systems and Software*, Vol. 68, No.3, pp. 253-262.

[10] Kotonya, G., and Sommerville, I., 1998, "Requirements Engineering: Processes and Techniques", Chichester, New York, John Wiley& Sons.

[11] Kvanli, A.H., Pavur, R. J., and Keeling, K. B., 2006,"Concise Managerial Statistics", Thomson Learning Inc.

[12] Keith, T., 2006, *Multiple regression and beyond.* PEARSON Allyn & Bacon.

[13] Kusuma Kumari B.M, 2014, "Software Cost Estimation Techniques", *International Journal of Emerging Research in Management &Technology*, vol.3, No.4, pp.104-108.

[14] Khatibi, V., and Jawawi, D.N.A., 2011, "Software Cost Estimation Methods: A Review", *Journal of Emerging Trends in Computing and Information Science*, Vol.2, No.1, pp.21-29.

[15] Lavazza, L., and Garavaglia. C., 2009, "Using function points to measure and estimate real-time and embedded software: Experiences and guidelines", *Proc. of the 3rd International Symposium on Empirical Software Engineering and Measurement*, pp.100-110.

[16] Lynch, J., 2009, Chaos manifesto. The Standish Group. [Online]. Available: http://www.standishgroup.com/newsroom/chaos_2009.php.

[17] Musilek, P., Pedrycz, W., Nan Sun and Succi, G., 2002, "On the sensitivity of COCOMO II software cost estimation model", *Proc. of the Eighth IEEE Symposium on Software Metrics*, pp.13-20**.**

[18] Montgomery, D. C., Peck, E. A., and Vining, G. G., 2006,"Introduction to Linear Regression", John Wiley & Sons.

[19] Nassif, A.B., 2012, "Software Size and Effort Estimation from Use Case Diagrams Using Regression and Soft Computing Models". *University of Western Ontario - Electronic Thesis and Dissertation Repository.* Paper 547.

[20] Shepperd M., and Schofield, C., 1997, "Estimating software project effort using analogies", *Software Engineering, IEEE Transactions on,* Vol. 23, pp. 736-743.

[21] Singh, Y., Bhatia, P.K., Kaur, A., and Sangwan, O., 2008, "A Review of Studies on Effort Estimation Techniques of Software Development", *Proc. Conference Mathematical Techniques: Emerging Paradigms for Electronics and IT Industries*, New Delhi, pp. 188-196.

[22] Shieh, G., 2010,On the misconception of multicollinearity in detection of moderating effects: Multicollinearity is not always detrimental. *Multivariate Behavioral Research,* Vol.45, pp.483-507.

[23] Sikka, G., Kaur, A., and Uddin, M., 2010, "Estimating function points: Using machine learning and regression models", *Proc. of the 2nd International Conference Education Technology and Computer (ICETC)*, Vol.3, pp.52-56.

[24] Suri P.K., and Pallavi Ranjan, 2012,"Comparative Analysis of Software Effort Estimation Techniques", *International Journal of Computer Applications*, Vol.48, pp.12-19.

[25] Tan, H. B. K., Zhao, Y., and Zhang, H., 2009, "Conceptual data model-based software size estimation for information systems", *ACM Transactions on Software Engineering and Methodology,* vol. 19, pp. 4:1-4:37.

[26] Xia, W., Capretz, L. F., Ho, D., and Ahmed, F., 2008, "A new calibration for Function Point complexity weights", *Information and Software Technology,* Vol.50, pp. 670-683.

[27] Yahya, M. A., Ahmad, R., and Lee, S. P., 2008, "Effects of Software Process Maturity on COCOMO II's Effort Estimation from CMMI Perspective", *Proc. of the IEEE International Conference on Research, Innovation and Vision for the Future,* RIVF. pp.255-262.

[28] Yinhuan, Z., Beizhan, W., Yilong Z., and Liang, S., 2009, "Estimation of software projects effort based on function point", *Proc. of the 4th International Conference on Computer Science & Education, ICCSE,* pp. 941-943.

