ISSN:0975-9646

Software Productivity Empirical Model for Early Estimation of Development

Mandeep Singh, Prof. Satwinder Singh
BBSBEC
Fatehgarh Sahib

Abstract-Software development efforts estimation is the process of predicting the most realistic use of effort required to develop or maintain software based on incomplete, uncertain and/or noisy input. Effort estimates may be used as input to project plans, iteration plans, budgets, investment analyses, pricing processes and bidding rounds.In this paper, analysis are performed on data to show the parameters which has maximum influence on the productivity. The model presented in this paper can be applied in any organization to calculate the influence of parameters on productivity.

Keywords-Productivity, Effort estimation, Duration.

1. INTRODUCTION

In the era of globalization, outsourcing, and fierce competition between different companies that produce software, the software development productivity issues are becoming increasingly important. High productivity largely facilitates, if not makes it possible, to quickly satisfy changing customer needs and still make money. Such abilities are what nowadays distinguish successful software companies from the less successful ones.

The productivity measure has become a tool for managers since it is used to compare the performance between different companies (benchmarking) and to compare the efficiency of different developers in the same company. Therefore, it allows doing strategic planning and decision making based on measurement. Many companies would like to benchmark the software development productivity of manpower for their projects[4]. If company has a well established metrics program and has a high project turnover, then it has a definite advantage[3]. One can benchmark the projects internally and avoid many of the measurement comparability problems associated with multi-company databases[4]. In addition, through the analysis of software project data, one can identify the factors that influence the productivity of the projects in the company. One may find that some of these important factors are given and unchangeable; for example, certain applications of a particular type may always be more difficult and associated with low productivity ratings. However, some variables such as choice of tool may be within your control.

2. LITERATURE REVIEW

Machek Ondrej [5] states that the definition and measurement of productivity is often inaccurate and differs from one method to another. Economic theory offers a well-grounded tool of productivity measurement. This article proposes a model of process productivity measurement based on the total factor productivity (TFP) approach commonly used in economics. It defines productivity and its measurement and also discusses the major data issues which have to be taken into consideration. Consequently, it apply the TFP approach in the domain of software engineering and we propose a TFP model of productivity assessment.

As per T Mukhopadhyay[1] Models are developed to estimate lines of code and function counts directly from user application features of process control systems early in the software development lifecycle. Since the application features are known with reasonable degree of confidence during early stages of development, it is possible to extend the use of the constructive cost model (COCOMO) and function-points-based approach for early software cost estimation. Alternative feature-based models that estimate size and effort using application features and productivity factors are developed. The feature-based models have been shown to estimate software effort with the least error.

K Maxwell[2] tells that identification, combination, and interaction of the many factors which influence software development productivity makes the measurement, estimation, comparison and tracking of productivity rates very difficult. Through the analysis of a European Space Agency database consisting of 99 software development projects from 37 companies in a European countries, the paper seeks to provide significant and useful Information about the major factors which influence the productivity of European space, military, and industrial applications, as well as to determine the best metric for measuring the productivity of these projects. The results indicate that some organizations are obtaining significantly higher productivity than others. Some of this variation is due to the differences in the application category and programming language of projects in each company; however, some differences must also be due to the ways in which these companies manage their software development projects. The use of tools and modern programming practices were found to be major controllable factors in productivity improvement.

As per Satwinder Singh[3] estimating software development effort is an important task in the management of large software projects. It is explored the use of Soft Computing Techniques to build a suitable model structure to utilize improved estimation of software effort for NASA software projects. A comparison between Artificial-Neural-Network Based Model (ANN) and Halstead, Walston-Felix, Bailey-Basili and Doty models were provided. The evaluation criteria are based upon MRE and MMRE. Consequently, the final results are very precise and reliable when they are applied to a real dataset in a software project. The results show that ANNs are effective in effort estimation.

Maxwell K D [4] examines a statistical analysis of a productivity variation, involving a unique database containing 206 business software projects from 26 Finnish companies. It examine differences in the factors, explaining productivity in the banking, insurance, manufacturing, wholesale/retail, and public administration sectors. The study provide productivity benchmarking equations that are useful both for estimating expected productivity at the start of a new project and for benchmarking a completed project for each business sector

Murat Yilmaz[6] state that approach can be useful for correlating latent (qualitative) variables and observable variables where empirical data can be collected. Consequently, the factors of interest can be revealed which aids managerial decision support. Second, we introduce the concept of social productivity and examined causal factors affecting productivity (leadership, team cohesion, collective outcome, trust) and identify their importance with respect to the opinion of our survey participants. Third, we introduce three variables to measure social capital of software development organizations (social relations, frequency of team meetings, interaction efficiency). Furthermore, we calculate several correlation values for factors investigated in both of our models. SEM is a modeling method frequently used to solve several problems encountered in social sciences. Our first structural model indicates that there is not only a significant amount of correlation between productivity and social productivity but also a correlation occurs among their interacting factors. Therefore in the refined model of productivity, we introduced social capital as a new latent variable and formalized our second model based on these facts. By modeling various aspects of productivity using a structural model, a researcher can obtain clear insights into the factors that are affecting productivity.

3. DATA SET

Data set cocomonasa/software cost estimation is taken from 60 NASA projects from different centers for projects from the 1980s and 1990s. Collected by Jairus Hihn, JPL, NASA, Manager SQIP Measurement & Benchmarking Element**.**

Sr No.	Variable Name	Description
1	acap	analysts capability
2	pcap	programmers capability
3	aexp	application experience
4	modp	modern programming practices
5	tool	use of software tools
6	vexp	virtual machine experience
7	lexp	language experience
8	sced	schedule constraint
9	stor	main memory constraint
10	data	data base size
11	syear	Starting year of project
12	time	time constraint for cpu
13	turn	turnaround time
14	virt	machine volatility
15	cplx	process complexity
16	rely	required software reliability

4. REGRESSION ANALYSIS

Regression analysis is a statistical process for estimating the relationships among variables. It is a statistical technique that allows us to predict impact on one variable on the basis of their scores on several other variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. Regression analysis helps to understand how the typical value of the dependent variable (criterion variable) changes when any one of the independent variables is varied, while the other independent variables are held fixed. A regression model shows *Y as* a function of X and β.

Y = f (X, β)

The β is denoted as unknown parameters, which may represent a scalar or a vector, X are independent variables and Y is the dependent variable

Coefficients Table

Model	Beta	t	Tolerance	VIF
(constant)	28.721	8.589		
Modp	-2.872	-2.832	.463	2.159
Stor	-3.778	-3.755	.880	1.137
Acap	-3.123	-3.573	.435	2.299
Data	-3.364	-4.440	.760	1.316
Cplx	-6.003	-3.977	.672	1.489
Pcap	-4.913	-2.331	.816	1.225
Turn	-6.751	-2.331	.543	1.841

The **Coefficients** table provides the details of the results. Both the raw and standardized regression coefficients are readjusted at each step to reflect the additional variables in the model. Ordinarily, although it is interesting to observe the dynamic changes taking place, we are usually interested in the final model. The Standardized Beta Coefficients give a measure of the contribution of each variable to the model. A large value indicates that a unit change in this predictor variable has a large effect on the criterion variable.

The beta value is a measure of how strongly each predictor variable influences the criterion variable. The beta is measured in units of standard deviation. For example, a beta value of 2.5 indicates that a change of one standard deviation in the predictor variable will result in a change of 2.5 standard deviations in the criterion variable. Thus, the higher the beta value the greater the impact of the predictor variable on the criterion variable.

The t and Sig (p) values give a rough indication of the impact of each predictor variable – a big absolute t value and small p value suggests that a predictor variable is having a large impact on the criterion variable. The value of t in our model is effective to show the impact of predictor variable on criterion variable.

The tolerance values are a measure of the correlation between the predictor variables and can vary between 0 and 1. The closer to zero the tolerance value is for a variable, the stronger the relationship between this and the other predictor variables. You should worry about variables that have a very low tolerance. SPSS will not include a

predictor variable in a model if it has a tolerance of less that 0.0001. However, you may want to set your own criteria rather higher – perhaps excluding any variable that has a tolerance level of less than 0.01. VIF is an alternative measure of collinearity in which a large value indicates a strong relationship between predictor variables. The value of tolerance and VIH in our model are effective which shows the strong relationship between predictor variables and criterion variable.

Y= -28.721-2.872(modp)-3.778(stor)-3.123(acap)-3.364(data)-6.003(cplx)-4.913(pcap)-6.751(turn)

Using this equation, given values for "modp" "stor," ,"acap", "data", "cplx", "cpacp"and "turn," you can come up with a prediction for the value of productivity of the project.

Model Summary

Model	R Square	Adjusted R Square
1	.3100	.299
2	.434	.415
3	.508	.483
4	.574	.545
5	.610	.576
6	.659	.622
7	.685	.645
8	.675	.640
9	.660	.631
10	.697	.664
11	.724	.689

R Square (R2) is the square of this measure of correlation and indicates the proportion of the variance in the criterion variable which is accounted for by our model. In essence, this is a measure of how good a prediction of the criterion variable we can make by knowing the predictor variables. However, R square tends to somewhat over-estimate the success of the model when applied to the real world, so an Adjusted R Square value is calculated which takes into account the number of variables in the model and the number of observations (participants) our model is based on. This Adjusted R Square value gives the most useful measure of the success of our model

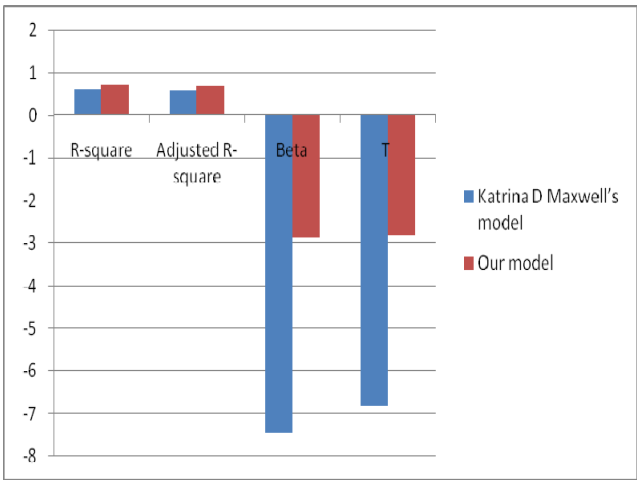
The Adjusted R Square value tells us that our model accounts for 68.9% of variance in the production score thus a very good model.

5. COMPARISON

The model is compared with the model of Katrina D Maxwell's model.

Parameter for comparison	**Katrina D Maxwell's model**	**Our model**
R-square	.613	.724
Adjusted R-square	.579	.689
Beta	-7.449	-2.872
T	-6.820	-2.832

Katrina D Maxwell's model which accounts 58% of variance where our model accounts 68.9% variance. The variance shows that our model is describing the variables having more influence on the production. Beta value is also greater than maxwell's model which show that there is more impact on dependent variable. Even the values of t are also improved from Maxwell's model. So with this model productivity can be improved more effectively.



The graphical representation of comparison between two models is shown in above graph. The above graph showing the improvement in the various values over the Katrina D maxwell's model.

6. CONCLUSION

Software productivity is an important process quality attribute. The model presented is based on cocomonasa/software cost estimation database. It has been shown how the model can be used to control the soft factors to optimize a particular project with respect to software productivity. The model discussed show the influence of particular variables on the productivity which means productivity can be improved by changing the variable in particular ratio. The Y value in our model shows how productivity is influenced by other variables.

REFERENCES

[1] T. Mukhopadhyay and S. Kekre, "Software Effort Models for Early Estimation of Process Control Applications," *IEEE Trans. Software Engineering*, Vol. 18, No. 10, Oct. 1992, pp. 915–924.

[2] K. Maxwell, L. Van Wassenhove, and S. Dutta, "Software Development Productivity of European Space, Military and Industrial Applications," *IEEE Trans. Software Eng.*, Vol. 22, No. 10, Oct. 1996, pp.706–718.

[3] Jaswinder Kaur, Satwinder Singh, Dr. Karanjeet Singh Kahlon, Pourush Bassi " Neural Network-A Novel Technique for Software Effort Estimation" International Journal of Computer Theory and Engineering, Vol. 2, No. 1 February, 2010 1793-8201

[4] Maxwell, K.D. Datamax, Avon "Benchmarking software development productivity" Volume: 17 , Issue: 1 Page(s): 80 – 88 Jan/Feb 2000

[5] Machek Ondrej, Hnilica Jiri and Hejda Jan " Estimating Productivity of Software Development Using the Total Factor Productivity Approach" 27 August 2012

[6] Murat Yilmaz and Rory V. O'Connor "An Empirical Investigation into Social Productivity of a Software Process: An approach by using the structural equation modeling" Communications in Computer and Information Science Volume 172, 2011, pp 155-166

[7] Rajiv D. Banker Sandra A. Slaughter " Project size and software maintenance productivity: empirical evidence on economies of scale in software maintenance " European Journal of Operations Research, Volume 17, Number I, July 1984,pp. 35-44

[8] Maxwell K.D., Applied statistics for Software Managers,Prentice-Hall, Upper Saddle river,2002.

[9] K. Maxwell, L. Van Wassenhove and S. Dutta, "Performance Evaluation of General and Company Specific Models in Software Development Effort Estimation," Management Science, Vol. 45, No. 6, June 1999, pp. 787-803

[10] R. Nevalainen and H. Maki, Laturi-System Productivity Model Version 1.4, Tech. Report 30.3.1994, Information Technology Development Center, Helsinki, 1994.

[11] C. Jones, Applied Software Measurement: Assuring Productivity and Quality, McGraw-Hill, New York, 1991.

[12] B. Boehm and R. Turner, Management challenges to implementing agile processes in traditional development organizations. IEEE Software, 22 (2005), 30-40.

[13] D.R. Graham, Incremental development and delivery for large software systems. Software Prototyping and Evolutionary Development, IEE Colloquium on, (1992), 2/1-2/9.