SOFTWARE TESTING EFFORT ESTIMATION WITH COBB-DOUGLAS FUNCTION: A PRACTICAL APPLICATION

Shaik Nafeez Umar

Lead-industry consultant, Test Management Consultant, AppLabs – a CSC Company, Hyderabad, India
nshaik5@csc.com, nafishaik123@gmail.com

Abstract

Effort estimation is one of the critical challenges in Software Testing Life Cycle (STLC). It is the basis for the project's effort estimation, planning, scheduling and budget planning. This paper illustrates model with an objective to depict the accuracy and bias variation of an organization's estimates of software testing effort through Cobb-Douglas function (CDF). Data variables selected for building the model were believed to be vital and have significant impact on the accuracy of estimates. Data gathered for the completed projects in the organization for about 13 releases.

Statistically, all variables in this model were statistically significant at p<0.05 and p<0.01 levels. The Cobb-Douglas function was selected and used for the software testing effort estimation. The results achieved with CDF were compared with the estimates provided by the area expert. The model's estimation figures are more accurate than the expert judgment. CDF has one of the appropriate techniques for estimating effort for software testing. CDF model accuracy is 93.42%.

Index Terms: *Effort estimation, software testing, Cobb-Douglas function, STLC and SDLC*

--***--

1. INTRODUCTION

Effort estimation in software industry is highly based on human judgment, Most of the IT organizations fail to deliver the projects on time. Most of the IT organizations use traditional and composite techniques effort estimation. There are different models available in market that is discussed below in brief.

Singh, Bhatia et al. [1] gave a more detailed classification of effort estimation empirical techniques, model/theory, expertise techniques, regression techniques, composite techniques and machine learning languages.

- Empirical techniques correspond to the analogy-based techniques which estimations are based on the practice and previous experience.
- Model/Theory based techniques are the algorithm based techniques that include Function Point Analysis, SLIM, Checkpoints and COCOMO model.
- Expertise techniques are equivalent to the expert judgment when a person carries out estimation based on non-explicit and non-recoverable reasoning [1];
- Regression based models are used to infer how the Y-variables are related to X-variable(s), requiring data from previous projects;
- Composite techniques combine both approaches - expert judgment and project data - in a consistent way in order to obtain the effort estimation [2, 3].

In the Software engineering literature there are many of the traditional models used to estimate the effort. Typically, the most common effort model used in testing field is COCOMO basic model [4, 5], SEL model, Walston –Felix model [6, 5], Bailey-Basil model [7], Halstead model [8], Doty model. From the existing effort prediction models no one indentify the right model to use in test effort estimation. These are all just about prediction models.

Advantages

- Help to effort prediction for each release
- Help to plan for preventive actions for effort estimation
- Improves the overall quality of the software product
- Streamlining the SDLC/STLC
- Insure users next to the costs of field defect occurrence
- Determine the reliability

2. OBJECTIVES AND METHODOLOGY

The main objective of this paper is to predict test effort in different test phases like Requirements Analysis, Test case design, Test execution, Test Automation, Test Governance and Project management. The proposed Cobb-Douglas model was implemented in excel sheet. The foremost step is to identify independent parameters of the testing effort model. These parameters should have impact or any relationship between effort and their corresponding independent parameters.

Before deployment of CMMI practices, organizations estimates were based on the judgment of only one expert. But, this is not considered to be a valid method and also not validated statistically. Two possible solutions were identified to meet the requirements of CMMI. One of the methods was Delphi method [9] with participation of at least three algorithmic models for the testing effort prediction. The other solution was implementation of algorithm model for the effort estimation based on the historical data of organization [9].

Model specifications includes input parameters like number of requirements, number of test cases, complexity of the release, number of resources, domain experience, requirements, test case design, test execution, test automation, test governance and project management. It predicts individual effort of testing phases and total effort. This model has analyzed the data of 14 software releases with parameters that are derived from the project dataset. The next step is to normalize the data. Next, check the correlation between the parameters and multi-collinerity crisis within the independent parameters. Then transform the data into logarithmic using the method of least squares to find coefficient values. Eventually, multiple linear regressions is used. The Cobb-Douglas function is given by

$$Y= \beta0+X1\beta1+ X2\ \beta2+ X3\ \beta3+ \ldots\ldots\ldots\ldots+ Xn\ \beta n$$

Where Y=Effort, β0=Constant, β1, β2, β3…… βn are coefficient values and X1, X2, X3…, Xn are independent parameters.

Accuracy of the model

In statistical modeling, accuracy is the one of the important criteria for measuring the performance of the model. After building the model, performance and precision of the model is evaluated. In this model, we used Theil's statistics to find out model performance. The Theil's statistical values lies between 0 and 1, the accuracy of the model is determined based on how close the value is to 1.

3. RESULTS AND DISCUSSION

Due to organization security and policy names of the projects are not disclosed. The Table#1 shows the distribution of release wise test data of different testing phases collected for 13 releases. The test data is depicted for both dependent and independent parameters.

The most significant R-squares (Table#2) of the dependent parameters of the model are Requirements (0.66), Test case design (0.75), Automated testing (0.68), Test governance (0.75) and Project management (0.85) which is highly statistically significant at 0.01 levels. This shows the selected dependent parameters selected for this model has significant impact on estimating the effort for software testing.

The standard error is comparatively less for requirement (6.58%), test case design (7.2%) and automated testing (5.69%). This shows statistically error of the model is very less for these variables when compared to other parameters. This indicates that these parameters are reliable and has significant impact in estimating the effort for software testing. In software testing organization most of the parameters are not associated with the effort due to frequent changes in the business requirements. Apparently, test execution is not statistically significant in model, it may be poorly correlated with effort and also its standard error is 11.52%, which is high in the model. The proposed Cobb-Douglas model gave 93.42% of accuracy of the actual and predicted values.

The variation between actual and predicted values using this model is significantly very low. When comparison of each every independent Cobb-Douglas model for testing phase it was good prediction actual vs predicted values. It was shows (Table#4) predicted values close to actual values. Finally in Cobb-Douglas model putting the actual independent parameters we will get predicted effort values.

Table1: Distribution of release wise data of different testing phases

Independent parameters						Actual effort (Hrs)-Dependent parameters						
Release	Number of requirements (#)	Number of test cases (#)	Complexity of the release (Scale)	Number of resources (#)	Domain experience (#)	Requirements	Test case design	Test execution	Automated testing	Test governance	Project management	Total effort
1	15	420	3	3	2	16	32	13	48	46	21	176
2	62	1240	2	5	2	15	43	14	52	56	20	200
3	65	1560	1	4	3	21	38	16	45	47	26	193
4	26	728	2	3	4	21	39	14	50	45	20	189
5	45	810	3	2	2	19	39	21	42	33	23	177
6	14	210	2	2	2	23	31	16	47	30	34	181
7	15	525	1	1	5	16	23	18	43	28	29	157
8	65	2600	2	8	6	23	25	16	37	28	24	153
9	48	1344	3	8	4	16	18	15	38	25	35	147
10	52	520	1	2	3	15	16	10	39	28	20	128
11(Assumed)	100	1800	3	4	5							
12(Assumed)	75	2400	4	6	4							
13(Assumed)	50	1250	2	4	3							

Table2: Descriptive statistics and R-square values of the testing parameters

Testing phases	Mean	Standard deviation	R-square	Standard error (%)
Requirements (Hrs)	18.50	3.27	0.66 (p<0.01)	6.58
Test case design (Hrs)	30.40	9.50	0.75 (p<0.01)	7.2
Test execution (Hrs)	15.30	2.95	0.48 (p>0.05)	11.52
Automated testing (Hrs)	44.10	5.17	0.68 (p<0.05)	5.69
Test governance (Hrs)	36.60	10.83	0.75 (p<0.01)	14.59
Project managemen (Hrs)	25.20	5.71	0.85 (p<0.01)	12.59

Table3: Predicted Effort (Hrs) of testing phases-using Cobb-Douglass function

Predicted effort (Hrs)-Testing phases							
Release	Requirements	Test case design	Test execution	Automated testing	Test governance	Product/ Proje management	Total effort
1	19	40	16	51	45	51	221
2	18	41	15	47	50	47	218
3	18	36	14	47	50	47	212
4	19	25	15	42	30	42	172
5	17	38	19	43	36	43	196
6	18	26	13	48	34	48	187
7	18	27	18	44	31	44	182
8	20	24	15	39	30	39	166
9	19	23	14	40	29	40	166
10	17	19	12	40	27	40	154
11(Assumed)	18	17	17	32	18	32	134
12(Assumed)	18	29	19	38	28	38	170
13(Assumed)	18	33	16	44	38	44	192

CONCLUSIONS

Based on the objective of the study Cobb-Douglas testing effort estimation model proposed is statistically and scientifically significant. Using the model the accuracy of effort estimation calculated is93.42%.It clearly states Cobb-Douglas model is more robust than other models as the model is blend of using historical data and scientific approach. With this level of accuracy, this model brings in huge relief for the software industry by helping them in estimating the testing efforts required for executing the project and thereby increasing the accuracy levels of the overall project effort estimates..

ACKNOWLEDGEMENTS

I want to thank Narendra singh Rajput, Lead Associate Manager, for his valuable feedback and guidance and Prof Balasiddamuni, Department of Statistics, S V University, Tirupati and Raghav Beeram, Global Head - Test Management Consulting, Independent Testing Services, CSC, Hyderabad for guiding me and supporting me in my research paper.

I would like to express my love affection to my mother Smt. Shaik Zaibunnisa, My Father Sri. Late S. Mahaboob Basha, Brothers and Sisters, My spouse Ballary Ameen, Children's Shaik Mohammed Wafeeq, Shaik Waneeya, my Niece Samreen Aaleia for enthusing me throughout the work

REFERENCES

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

[2]. B. Boehm, C. Abts and S. Chulani, Software development cost estimation approaches - A survey, Annal of Software Engineering, vol. 10, pp. 177-205, 2000.

[3]. L.C. Briand and I. Wieczorek, Resource Estimation in Software Engineering, Encyclopedia of Software Eng., J.J. Marcinak, ed., pp. 1160-1196, John Wiley & Sons, 2002.

[4]. M.V. Deshpande and S.G. Bhirud, "Analysis of Combining Software Estimation Techniques," International Journal of Computer Applications (0975 – 8887) Volume 5 – No.3.

[5]. Y. Singh, K.K.Aggarwal. Software Engineering Third edition, New Age International Publisher Limited New Delhi.

[6]. Pressman. Software Engineering - a Practitioner's Approach. 6th Eddition Mc Graw Hill international Edition, Pearson education, ISBN 007 - 124083 – 7.

[7]. S. Devnani-Chulani, "Bayesian Analysis of Software Cost and Quality Models”. Faculty of the Graduate School, University Of Southern California May 1999.

[8]. O. Benediktsson and D. Dalcher, “Effort Estimation in incremental Software Development,” IEEE Proc. Software, Vol. 150, no. 6, December 2003, pp. 351-

[9] H. Leung and Z. Fan, Software Cost Estimation, in Handbook of Software Engineering and Knowledge Engineering, Hong Kong Polytechnic University, 2002

BIOGRAPHIES

Working as Lead Industry Consultant in Test Management Consulting group, AppLabs – a Computer Science Corporation Company, Hyderabad, India. He has 9 years in IT experience as Statistician and Statistical modeler. He has published 14 papers in National/International journals. He is expert in Statistical and Econometric modeling