2013 39th Euromicro Conference Series on Software Engineering and Advanced Applications

Accuracy of Contemporary Parametric Software Estimation Models: A Comparative Analysis

Derya Toka
Ads Quality Operations
Google Ireland Ltd.
Dublin, Ireland
deryat@google.com

Oktay Turetken
Information Systems Group, School of Industrial Engr.
Eindhoven University of Technology
Eindhoven, Netherlands
o.turetken@tue.nl

***Abstract*— Predicting the effort, duration and cost required to develop and maintain a software system is crucial in IT project management. Although an accurate estimation is invaluable for the success of an IT development project, it often proves difficult to attain. This paper presents an empirical evaluation of four parametric software estimation models, namely COCOMO II, SEER-SEM, SLIM, and TruePlanning, in terms of their project effort and duration prediction accuracy. Using real project data from 51 software development projects, we evaluated the capabilities of the models by comparing the predictions with the actual effort and duration values. The study showed that the estimation capabilities of the models investigated are on a par in accuracy, while there is still significant room for improvement in order to better address the prediction challenges faced in practice.**

Keywords*— *project estimation; effort estimation; parametric estimation model; software estimation; COCOMO II; SLIM; SEER-SEM; TruePlanning; Magnitude of Relative Error (MRE)

I. INTRODUCTION

Despite the increasing need for software systems in all aspects of human life, software development projects are historically notorious with delays and being costly and error-prone. Inaccurate estimation of project resources is considered as one of the top project failure factors [1]. As such, software estimation has continued to attract attention from academicians and practitioners. Project managers are often faced with the myriad challenges of estimating the costs, schedule and the resources for IT projects.

Software estimation is the process of predicting the effort, duration and cost required to develop a software system [2]. Estimators often rely on one or more rules-of-thumb to create software estimations [3]. Underestimating the costs may result in exceeded budget, underdeveloped functions, poor quality, and failure to complete on time while overestimating may end with too many resources committed to the project [2].

The pressure to present accurate estimates has led to the development of specific practice of methods for IT projects in recent years. The parametric models are those that apply a set of mathematical equations, and typically incorporate diverse properties of project or context as variable parameters [4]. Despite the fact that these estimation models have existed for many years, the performance of these models in terms of prediction accuracy has constantly been debated [5], [6].

The main objective of this study is therefore to benchmark and report on the performance of a set of commonly used parametric estimation models in terms of their accuracy in estimating IT project effort and duration by applying the selected estimation models. Consequently, the main question for this research study is: *How accurate are the contemporary parametric effort and duration estimation models?* For this purpose, we selected four widely used parametric software estimation models and apply them to predict the effort and duration of 51 completed real-life projects. We performed a comparative analysis on the estimates obtained from the application of these models and the actual effort and duration values of the selected projects.

II. RELATED WORK

Several approaches have been developed for estimating the effort and duration of IT projects. Expert judgement (such as the Delphi method), analogy-based, parametric models (which include regression-based or model-based techniques), simulation and neural network approaches are those that are commonly referred in research and practice [7].

Over the last three decades, there has been an increasing trend on the research performed particularly on the parametric estimation models [5] as they are considered to be relatively accurate compared to other estimation methods [8]. Several models were developed that are coupled with a tool that implements the technique to help for the prediction. COCOMO II, SEER-SEM, QSM-SLIM, TruePlanning, KnowledgePlan, SoftCost, ESTIMACS, CHECKPOINT, and CostXpert are the methods/tools commonly referred in practice [9], [8].

However, despite the extensive research on developing new and better software estimation models, very few studies exist on the analysis of the performance of widely used parametric software estimation models. Mittas et al. [6] proposes the use of statistical simulation tools to compare the accuracy of two techniques: the estimation by analogy and the regression analysis. Berlin et al. [10] reports on a similar study that contrasts two types of models: linear regression models and models that are based on artificial neural networks. The analysis by Briand et al. [11] compares also regression-based and analogy-based models.

Basha et al. [12] provide a literature-based comparison of several empirical software effort estimation models. Jensen et al. [13] provide a description of specific parametric software

978-0-7695-5091-6/13 $26.00 © 2013 IEEE
DOI 10.1109/SEAA.2013.49

cost estimation models, namely SEER-SEM, SLIM, and CostXpert, and evaluate them by looking at the formulation and viewpoint underlying each method. However, these works do not provide any comparison in terms of the prediction accuracy of these models.

Kemerer's work [14] is one of the first to report on the quantitative evaluation of early parametric estimation models, namely SLIM, COCOMO 81, Function Points and ESTIMACS. Similarly, Stark [3] presents a comparative analysis of COCOMO 81, Jones, ISBSG, and Rone estimation models. The works both by Kemerer [14] and Stark [3] share a common purpose with the study presented in this paper. However, they take the early (and mostly obsolete) estimation models, such as COCOMO 81, Rone, under consideration.

III. Research Method

The method that we followed for our research on the analysis of the models is depicted in Fig. 1.

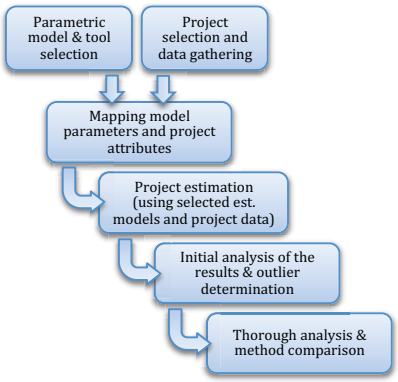

Fig. 1. Research Method – Workflow

A. Parametric Model and Tool Selection

After reviewing several empirical parametric estimation models and tools, the following four models (and corresponding tools) were selected for the comparative analysis: Boehm's COCOMO II, Galorath's SEER-SEM, QSM's SLIM-Estimate (SLIM) and Price-S' TruePlanning. The selection is mainly due their widely use and adoption in practice [15], [9], [8], [12] and their applicability in estimating the development of business applications. Three of these tools are commercial (SEER-SEM, TruePlanning, and SLIM) and they responded to our academic license request, whereas COCOMO II provides a free-online tool for estimations.

- *COCOMO II by BOEHM:* COCOMO (COnstructive COst MOdel) is the most well-known parametric software cost estimation models, which was developed by Barry W. Boehm in 1981 [16]. COCOMO II is the latest major extension to the original model [17]. It also serves as the base model in many other commonly used commercial estimation tools such as SoftStar Systems' Costar (http://www.softstarsystems.com/).
- *SEER-SEM by GALORATH*: SEER (System Evaluation and Estimation of Resources) is a proprietary project estimation model owned by Galorath International Ltd. Co. Galorath which was established in 1979 [8].
- *SLIM by QSM*: SLIM (Software Life-Cycle Model) is a software estimation model developed in the late 1970s by Larry Putnam, who is the president of Qualitative Software Management (QSM) Inc. [18].
- *TruePlanning by PRICE Systems:* PRICE-S is a model developed by PRICE Systems Inc. TruePlanning is the latest version of PRICE-S (www.pricesystems.com).

Training sessions were offered to the authors by Galorath, QSM Europe, and Price Systems. As COCOMO II is available as a free-online tool, manuals regarding the tool and model usage were freely available.

B. Project Selection and Data Gathering

There are several software project data repositories (software benchmark data sources) that offer project data collected by companies, non-profit organizations, research groups, and universities. Examples of these sources include; International Software Benchmark Standards Group (ISBSG), Capers Jones & Associates LLC, Galorath Incorporated, QSM Inc., and Q/P Management Group, Inc. [19].

We selected ISBSG (International Software Benchmarking Standards Group) repository as the source for the project data for our analysis as it is internationally recognized and independent organization. The version we used (Release 11, 2009) has data from over 5000 international projects related to software development and enhancement [20].

The following criteria were applied for the selection of sample projects from the repository:

- *Development Type*: Only 'New development' projects were selected. 'Enhancement' projects were excluded as this type of projects may involve unconventional structures with typical development lifecycle activities missing or not applicable.
- *Year and Status*: Only 'completed' projects that are implemented since 1997 were included in the list.
- *Data Quality*: Data Quality is one of the ISBSG data attribute which shows the quality of the project data with 4 scales: A, B, C, and D (A being the most accurate data). Only those projects that have 'A' data quality rating were selected in order to maintain high reliability in the data.
- *Software Functional Size Counting Approach*: The software size is considered as one of the key inputs in parametric software estimation. IFPUG Function Point Analysis [21] is the most commonly used method in practice. Only those projects, that are counted using the IFPUG method (with version 4.1 or above), were selected to ensure consistency and avoid differences that may result in using different sizing approaches.
- *UFP (Unadjusted Function Point) Quality*: The UFP Quality field in the ISBSG repository represents the quality of the functional size measurement assessed by the ISBSG Quality Reviewers. The rating codes are A, B, C and D. Only those projects with rating 'A' were selected to ensure the quality of data to be analyzed.
- *IFPUG VAF (Value Adjustment Factor)*: The IFPUG VAF value for a software is calculated using 14 system characteristics that rate the general functionality of the application being counted [21]. The closer the VAF to 1.0, the more 'typical' the software is. We selected the projects that have VAF value between 0.90 and 1.10 in

order the decrease the variation among selected projects in terms of several system characteristics.

- *Software (Functional) Size*: The projects that had sizes between 100 and 1050 FPs are included in the data set.

In total, 56 projects in the ISBSG repository satisfied the criteria specified above. The selected set consisted of 'new development' projects with average size of 293 FPs (min. 113 FP, max. 1048 FPs). Additionally, the programming languages used for the selected projects are evenly distributed.

C. Mapping Models' Input Parameters to ISBSG Project Data Attributes

To contribute to a systematic and unbiased estimation based on the project data, a mapping scheme was developed between each estimation model input parameters and ISBSG project data attributes. This helped to decrease estimators' subjectivity in interpreting the project characteristics, and contributed to the consistency and repeatability of the estimations.

As each estimation model requires a specific set of input parameters, a separate mapping scheme was developed for four models (Table I). For those parameters that do not have any corresponding ISBSG project attribute, the values for the model parameters were assumed as 'nominal' by default.

TABLE I NUMBER OF MODEL INPUT PARAMETERS AND THEIR MAPPING TO ISBSG PROJECT ATTRIBUTES

Method	# of input param.	# of param. that could be mapped to ISBSG project attributes	Coverage (mapped/ total # of parm.)
COCOMO II	27	11	40%
SEER-SEM	50	30	60%
SLIM (with Quick Estimate option)	11	7	64%
TruePlanning	40	15	38%

D. Project Estimation

Based on the corresponding mapping schemes, the effort and duration values for each project were estimated by four estimation models by one of the authors. The estimation for the entire project set is independently repeated twice. The resulting numbers (for effort and duration) from the rounds are consolidated for each project and the model. In addition, estimations for each model for a set of randomly selected 15 projects were evaluated by the second author of this paper.

E. Initial Analysis of the Results and Outlier Determination

The estimated efforts and durations for 56 pre-selected projects for each estimation model were compared with the actual work effort and duration (project elapsed time). The following metrics were measured for each project and model:

- *MRE (Magnitude of Relative Error)* is one of the most commonly used metrics for measuring the reliability of an estimation [22].
  MRE = |Actual value − Estimated value| / |Actual value|
  MMRE (Mean MRE) indicates the average relative error of a set of estimates (e.g. of a particular method).

- *PE (Percentage Error)* is another similar method used to measure the accuracy of prediction [12]. The only difference between MRE and PE is that MRE calculates the absolute value.
  PE= Actual value − Estimated value / Actual value

- *Actual %* is the percentage of the actual value with respect to the estimate.
  Actual % = Actual value / Estimated value x 100

While MRE signs how accurate the estimation is, PE represents if the estimation is optimistic or pessimistic (i.e. if the project is overestimated or underestimated). The closer the MRE value to 0%, the more accurate the estimation is. Actual% is ideally equal to 100%, i.e. equal to the estimation.

As an initial step, the MRE, PE and Actual% values for 56 projects were calculated for each method. Due to their exceptionally high MRE values (greater than 500%), 5 projects were considered as *outliers* and were eliminated from the selected project list.

IV. RESULTS OF THE ANALYSIS

A. Effort Estimation Performance

Table II presents the average values (and standard deviations) for the effort estimations. Considering the Actual% values, SEER-SEM, SLIM and TruePlanning models can be considered to follow optimistic approaches in effort estimation. In other words, it is more likely that estimations by these models will be below the actual effort. COCOMO II, on the other hand, can be considered as pessimistic.

TABLE II AVERAGES REGARDING **EFFORT** ESTIMATES

Estimation Models	Average			Standard Deviation		
	Actual %	*MMRE*	*PE*	*Actual %*	*MMRE*	*PE*
COCOMO II	91%	74%	-54%	53%	80%	95%
SEER-SEM	112%	36%	-10%	55%	41%	54%
SLIM	119%	41%	-7%	59%	43%	60%
TruePlanning	116%	34%	-4%	52%	38%	51%

The results show that the MRE values for COCOMO II estimations are higher when compared to the others. In order to signify the conclusions, statistical T-Tests were conducted to compare the means (MMRE) with 95% confidence level. We performed six T-Tests in total (between each method) and the results showed that TruePlanning, SEER-SEM and SLIM provided better estimates than COCOMO II (with $p < 0.02$). However, the performance of these three models (namely, TruePlanning, SEER-SEM and SLIM) cannot be statistically differentiated with respect to the effort prediction accuracy.

B. Duration Estimation Performance

Table III shows the overall performance of the methods in estimating duration. In that, all models can be considered to follow a pessimistic view as the average duration estimations are above the actual values.

TABLE III AVERAGES REGARDING **DURATION** (SCHEDULE) ESTIMATES

Estimation Models	Average			Standard Deviation		
	Actual %	*MMRE*	*PE*	*Actual %*	*MMRE*	*PE*
COCOMO II	77%	91%	-80%	*48%*	*92%*	*102%*
SEER-SEM	85%	81%	-65%	*64%*	*80%*	*93%*
SLIM	99%	84%	-60%	*77%*	*96%*	*113%*
TruePlanning	85%	99%	-82%	*61%*	*116%*	*129%*

As seen in the table, the duration estimation performances of the methods are comparable, although the deviations in the values by TruePlanning are relatively higher. According to the statistical analysis, although SEER-SEM scored the lowest average MRE (and corresponding standard deviation) values, the T-Tests indicated that there is no sufficient evidence to argue any difference between the methods in terms of their duration estimation performances.

V. Discussions and Conclusions

This study aimed at investigating the estimation accuracy of four widely used parametric software estimation models by comparing their performances on 51 software development projects residing in the ISBSG project repository. As a summary, the results show that:

- With regard to effort estimation: The accuracy levels of TruePlanning, SEER-SEM and SLIM are alike. COCOMO II, on the other hand, scored the lowest in terms of effort estimation accuracy. The COCOMO II follows a pessimistic approach, while the approach followed by the other three is optimistic.
- With regard to duration estimation: Although SEER-SEM had the lowest MMRE value, all four methods are considered to yield equivalent estimates according to the statistical analysis, and pessimistic in estimating duration.
- COCOMO II performed better in estimating the project duration than the effort.
- SEER-SEM was (relatively) successful in both effort and duration estimation.
- TruePlanning performed better in estimating effort (than duration).

Our analysis suggests that the COCOMO II model is inferior to the other three models in estimating effort, whereas, we did not find enough evidence to suggest any differences in the performances of these four models in terms of estimating duration. However, the resulting MMRE values for all methods can be considered *high* in general. We believe that these levels of accuracy can be considered *inadequate* in practice. This also suggests significant improvement opportunities for these models and room for new models to better address the estimation challenges that the projects are facing today.

A. Limitations and Future Work

A major limitation for the study is the partial project information in the ISBSG project repository. As mentioned before, for some projects, data for some significant attributes were missing, which is likely to have influenced the accuracy level of the estimations. As a future work, the project list can be populated only with those projects that have some key project attributes present. Alternatively, the study can be replicated using other project data sources (including also from companies), with better-input parameter coverage.

Related to the limitation discussed above; the performance of these tools in terms of the accuracy level should also be interpreted with the fact that the commercial tools we analysed -SEER-SEM, SLIM, and TruePlanning- offer opportunities for calibrating the estimation process with historical project data. Therefore, future work can be considered as incorporating historical data for calibration purposes to have more insight into the capabilities and strengths of these methods and tools.

The number of software projects can be increased for more reliable results and conclusions. This may also allow for additional statistical analysis to investigate the relationship between effort and duration, and their effects on each other. Similarly, correlations between project variable and estimation accuracy can be more accurately and reliably investigated with more data.

The constraint on the size of the projects used in this study is another limitation that poses threats on the results. As a future work, projects with varying sizes and with a wider coverage – in terms of project size- can be studied.

Acknowledgment

The authors gratefully acknowledge Galorath Int. Ltd., QSM Inc., and PRICE Systems Inc. for their valuable contributions and effort in offering the resources for this study.

References

[1] R. N. Charette, "Why Software Fails?," *IEEE Spectrum, Sept.2005*, 2005.

[2] H. Leung and Z. Fan, "Software Cost Estimation," pp. 1–14, 2002.

[3] G. Stark, "A Comparison of Parametric Software Estimation Models Using Real Project Data," *Crosstalk*, no. Jan/Feb 2011, 2011.

[4] PMI, *A Guide to the Project Management Body of Knowledge, PMBOK Guide, 4th Ed.* 2008.

[5] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Transactions on Software Engineering*, vol. 33, no. 1, pp. 33–53, 2007.

[6] N. Mittas and L. Angelis, "Comparing cost prediction models by resampling techniques," *Journal of Systems and Software*, vol. 81, no. 5, pp. 616–632, May 2008.

[7] B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approaches – A survey," *Annals of Software Engineering*, vol. 10, pp. 177–205, 2000.

[8] M. W. E. Daniel D. Galorath, *Software Sizing, Estimation, and Risk Management*. Auerbach Publications, 2006.

[9] C. Jones, *Applied Software Measurement: Global Analysis of Productivity and Quality*. Mc Graw Hill, 2008.

[10] S. Berlin, T. Raz, C. Glezer, and M. Zviran, "Comparison of estimation methods of cost and duration in IT projects," *Information and Software Technology*, vol. 51, no. 4, pp. 738–748, Apr. 2009.

[11] L. C. Briand, K. El Emam, D. Surmann, I. Wieczorek, and K. D. Maxwell, "An assessment and comparison of common software cost estimation modeling techniques," in *ICSE '99*, 1999, pp. 313–322.

[12] S. Basha and P. Dhavachelvan, "Analysis of Empirical Software Effort Estimation Models," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 7, no. 3, pp. 68–77, 2010.

[13] R. Jensen, L. P. Sr, and W. Roetzheim, "Software estimating models: three viewpoints," *The Journal of Defense Software*, pp. 23–29, 2006.

[14] C. F. Kemerer, "An empirical validation of software cost estimation models," *Communications of the ACM*, vol. 30, no. 5, pp. 416–429.

[15] L. Buglione and C. Ebert, "Estimation Tools and Techniques," *IEEE Software, May/June 2011*, 2011.

[16] B. Boehm, *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[17] B. Boehm et al., *Software Cost Estimation with COCOMO II*. Englewood Cliffs, NJ: Prentice-Hall, 2000.

[18] Lawrence H. Putnam and Ware Myers, *Five Core Metrics*. Dorset House Publishing, 2003.

[19] C. Jones, "Sources of Software Benchmarks, Ver.19," Capers Jones & Associates LLC, 2012.

[20] ISBSG-Demographics, "Data Demographics Release 11 What you can find in the D & E Repository," pp. 1–24, 2009.

[21] IFPUG, "IFPUG (International Function Point Users Group) Function Point Counting Practices Manual, Release 4.3.1," 2010.

[22] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit, "A simulation study of the model evaluation criterion mmre," *IEEE Transactions on Software Engineering*, vol. 29, no. 11, pp. 985–995, Nov. 2003.