

Identifying Factors Affecting Software Development Cost

Robert Lagerström, Liv Marcks von Würtemberg, Hannes Holm and Oscar Luczak
Industrial Information and Control Systems
The Royal Institute of Technology
Stockholm, Sweden
Email: {robertl, livm, hannesh, oscarl}@ics.kth.se

Abstract—Software systems of today are often complex, making development costs difficult to estimate. This paper uses data from 50 projects performed at one of the largest banks in Sweden to identify factors that have an impact on software development cost.

Correlation analysis of the relationship between factor states and project costs were assessed using ANOVA and regression analysis.

Ten out of the original 32 factors turned out to have an impact on software development project cost at the Swedish bank, including the number of function points and involved risk.

Some of the factors found to have an impact on cost are already included in estimation models such as COCOMO II and SEER-SEM, for instance function points and software platform. Thus, this paper validates these well-known factors for cost estimation.

However, several of the factors found in this study are not included in established models for software development cost estimation. Thus, this paper also provides indications for possible extensions of these models.

Keywords—cost estimation; software development cost; estimation models; factors; function points

I. INTRODUCTION

A great majority of today's companies are highly dependent of software systems supporting their business processes. When the business processes are modified to meet new requirements, the software systems have to undergo a corresponding change or develop new supporting software. This is also the case when a company introduces new business processes.

As software systems of today often are interconnected with each other, developing a new system frequently involve modifications of old systems as well. This makes software development costs difficult to predict and methods for development cost estimation useful as decision support. The development of completely new software systems, as well as the modifications of old systems, are often performed in projects. [1] ascertain that 23% of all software projects are cancelled before completion. Furthermore, of the completed projects, only 28% are delivered on time with the average project overrunning the budget with 45% [1]. Despite the existence of several established cost estimation models, the Standish Group reports that this situation has aggravated during the last couple of years [2].

Apparently, the need for a well reinforced decision basis regarding cost estimation is as great as ever. Therefore, the research question addressed in this paper is: What factors have an effect on software development project costs?

This paper analyzes data from 50 projects performed at one of the largest banks in Sweden. In total 32 factors are studied. Using one-way ANOVA and bivariate regression analysis, ten factors that significantly affect project costs are identified. Several of the factors found are not included in established models for software development cost estimation, such as COCOMO II and SEER-SEM. The contribution of this paper is therefore proposed extensions to established cost estimation models. Also, some of the factors studied are already used in the existing models. Thus, some of these factors are further validated by this study and some are challenged.

A. Outline

The remainder of the paper is structured as follows: Section II describes different methods used for software development cost estimation. The factors included in the analysis are listed in section III, whereas the statistical methods used in the analysis are described in section IV. Results from the analysis are presented in section V and further discussed in section VI. Finally, section VII summarizes the paper with conclusions.

II. RELATED WORK

Many different techniques for software development cost estimation have been elaborated by numerous researchers, this chapter briefly presents some of these methods with focus on factor-based models. As a majority of the established models use function points [3] and function points are a part of the study presented in this paper, the factor is addressed separately in section II-A.

A. Function points

Function point analysis is a way of measuring the size and extent of a software system by looking at which functions the system delivers to the user. Function points was first introduced by Albrecht and Gaffney in 1983 [4] and it has since then been improved and tested in many studies. Today people often use the function points based method called

COSMIC [5]. If used correctly, function points is considered to be of considerable assistance when estimating software development costs. However, counting function points can be resource-demanding and difficult. A good result also requires accurate approximations in the subsequent regression analysis [6].

B. Factor-based models

The COConstructive COst MOdel (COCOMO), originally published by [7] and upgraded to COCOMO II in 1995 [3], uses probabilistic Bayesian networks to turn accessible data into an estimated cost [8]. Factors considered by COCOMO II include size attributes such as number of lines of source code or function points and additional cost drivers required, each related to one of four categories: 1) Platform - software reliability, data base size, required reusability, documentation match to life-cycle needs and product complexity. 2) Product - execution time constraint, main storage constraint and platform volatility. 3) Personnel - analyst capability, programmer capability, personnel continuity, applications experience, platform experience and language and tool experience. 4) Project - project use of software tools, multisite development, required development schedule, precedentedness, development flexibility, architecture/risk resolution, team cohesion and process maturity [3].

A newly developed method called The Enterprise Architecture Modifiability Analysis Tool (TEAMATe) also uses probabilistic Bayesian networks, with the additional feature of being coupled with enterprise architecture models. TEAMATe consists of the following factors: Change management process maturity, documentation quality, software system understandability, software system size, software system internal and external coupling, software system change size, software system change difficulty, quality of tools for software system changes, quality of infrastructure for software system changes, project team expertise, project members time on project, number of project members and software system change activity synchronization need [9], [10].

System Evaluation and Estimation of Resources - Software Estimation Model (SEER-SEM) is offered by Galorath Inc. of El Segundo, California [11]. SEER-SEM, based on the Jensen model first published in 1983 [3], takes a parametric approach to estimation. By stating parameters such as size, personnel, complexity, environment and constraints the model submit estimations of for instance project cost, risk and schedule. SEER-SEM covers all phases of a project life-cycle and handle several environmental and application configurations [3], [8].

In [3], [8] several other more or less well-known factor-based models are presented and discussed such as PRICE-S [12], ESTIMACS [13], Checkpoint [14], Softcost [15], the Putnam Software Life Cycle Model (SLIM) [16], the Jensen Model [17] and the Bailey-Basili Model [18]. Furthermore,

Jorgensen presents a systematic review of software development cost estimation studies in [19].

C. Dynamic models

Dynamic-based techniques pay regard to circumstances that change during a project, including changes that have a non-linear influence on the overall cost. Examples of these kinds of circumstances are that project personnel often are more efficient when they have been working in a project for a long while, or that extra personnel in a project increases the time spent on coordination.

System dynamics simulation models use sets of first-order differential equations to estimate software development costs [3].

D. Expert-based methods

Expertise-based techniques use knowledge and experience from participants in former projects rather than empirical data [3]. Two common examples of expertise-based techniques are: 1) WBS (Work Breakdown Structure), where the products and processes respectively are divided into more lucid parts [3], [7]. 2) Planning Poker, where project team members first estimate the cost separately and then discuss the result until consensus is obtained [20].

III. STUDIED PROJECTS & FACTORS

A pre-study consisting of studying project documentation at one of the largest banks in Sweden was employed in order to assemble a list of factors having potential impact on software development cost. All project data used in the analysis has been found in the project database at the bank. The study only includes development and integration projects involving more than 200 man-hours and/or 500 000 SEK. Further, 7.28 SEK equals 1.00 USD and 10.11 SEK equals 1.00 EUR (2010.02.04). The data samples, consisting of data from 50 projects, include all projects fulfilling the above stated criterias that was performed at the bank from the summer 2006 until the summer 2008. Therefore the data can be said to be representative, at least for the bank in question.

In the data collection phase, plans, follow-up documentation and scorecards for each project were used. All factors included in this study comes from the internal documentation performed continuously at the bank following the normal procedures for executing projects. This study is thus delimited to factors documented by the bank.

The response variable used in the study, software development project cost, was measured in Swedish 'kronor' (SEK), the currency used in Sweden.

A total of 32 factors were identified during the pre-study. These factors and their corresponding states are presented in this section of the paper according to the following syntax:

Factor name: *Factor states*
Factor description.

Evaluated factors:

Function points: *Real numbers ≥ 0*

Number of function points in a project.

Primary platform: *DW / TDE / ZOS / other*

Main software platform affected by a project. Possible states are DataWarehouse (DW), Touchdown Europe (TDE) and a 64-bit operating system for mainframes (ZOS).

Secondary platform: *DW / TDE / ZOS / other*

Secondary software platform affected by a project.

Presentation interface: *3270 / Web browser / Windows / None*

Main interface of the delivered software in a project.

Risk classification: *High / Medium / Low*

Overall assessed risk associated with a project.

Existence of overall schedule: *Yes / No*

If a project has included an overall plan or not.

Existence of overall testing plan: *Yes / No*

If a project has included an overall plan for testings or not.

Existence of testing conductor: *Yes / No*

If a project has included a testing conductor or not.

Length of pre-study: *Real numbers ≥ 0*

Length of a project's pre-study phase, measured as a fraction of the total time spent on a project.

Cost of pre-study: *Real numbers ≥ 0*

Cost of a project's pre-study phase, measured as a fraction of the total cost spent on a project.

Project type: *Development / Integration*

Type of project being carried out.

Project priority: *High priority / Low priority*

Internal prioritization of a project.

Business manager: *Name*

Individual responsible as business manager in a project.

Project manager: *Name*

Individual responsible for managing a project.

Liabe for delivery: *Name*

Individual responsible for the delivery of results in a project.

Architect: *Name*

Individual assigned the role as system architect in a project.

Final deadline revisions: *Natural numbers*

Number of changes that have been made to a project's final deadline.

Budget revisions: *Natural numbers*

Number of budget revisions during a project.

Method for debit: *Continuous / Fixed / Manual*

How the method for debit has been handled in a project.

Project participants: *Natural numbers*

Number of individuals who have spent more than 40 hours within a project.

Duration: *Domain of natural numbers*

Total number of workdays a project have utilized.

External parts: *Natural numbers*

Number of external units involved in a project.

Consultants: *Natural numbers*

Total number of consultants participating in a project.

Cooperation: *1 (very poorly) - 5 (very well)*

How well involved project participants cooperated.

Integrations testing: *Yes / No*

If integrations testing has been performed or not in a project.

Commissioning body: *Name*

Project sponsor.

Commissioning unit: *One of nine units*

Unit cohering to respective commissioning body.

Competence performing assignment: *1 (very low) - 5 (very high)*

How the commissioning body assessed the competence of an involved project group.

Performance of estimation- and prognosis efforts: *1 (very poorly) - 5 (very well)*

How the commissioning body assessed the performance of estimation- and prognosis efforts.

Quality of delivery: *1 (very low) - 5 (very high)*

How the commissioning body assessed the quality of the delivery.

Conformance to requirements: *1 (very low) - 5 (very high)*

How the commissioning body assessed the final delivery compared to original requirements in a project.

Implementation efficiency: *1 (very low) - 5 (very high)*

How the commissioning body assessed the efficiency of implementation.

IV. ANALYSIS METHOD

Each set of attributes connected to the factors described in section III belong to one of four different measurement

scales; either the nominal, ordinal, interval or ratio scale. Inherent properties of involved ranges generate a need for different statistical tools when assessing impact on cost from involved factors.

Factors belonging to nominal or ordinal scales are through reasons described in [21] assessed using one-way between-subjects analysis of variance and factors belonging to interval or ratio scales by recommendation of [21] using a bivariate regression analysis.

A. One-Way Between-Subjects Analysis of Variance

One-Way Between-Subjects Analysis of Variance, or one-way ANOVA, is used in situations where the purpose is to compare means of a quantitative Y outcome variable across two or more groups [21]. Furthermore, 'One-Way' implies that there is only one factor involved in the study.

One of the key assumptions of one-way ANOVA, other than quantitative measurements, is that assessed scores are normally distributed across the entire sample and group, with no extreme outliers [21]. If data seem to cohere to another distribution than the normal distribution it might be possible to remedy the problem through a transformation, or by removing outliers [21].

The null hypothesis of an ANOVA-test, H_0 , is true if the differences between observed groups of data can be described by chance and false if there are systematic differences large enough to justify rejection of H_0 [21]. The boundary associated with rejecting a null hypothesis is generally described using probability, p . Datasets which contain groups of observed data with differences large enough to reject the null hypothesis are statistically significant. A commonly used probabilistic boundary is $p < 0.05$, which implies that there is less than 5% probability for H_0 to be true [21]. The reader is referred to the following work regarding information about the technical aspects for identifying actual boundaries; [21], [22].

B. Bivariate regression

A bivariate regression analysis provides a linear equation that models the expected outcome on a quantitative variable Y from data on a variable X [21], mathematically defined as

$$Y' = b_0 + bX.$$

Where Y' is the predicted outcome variable, b_0 is the intercept, and b is the slope. The regression coefficient b and the intercept b_0 are further identified using mathematical techniques found in [21], [22] to reach optimal fitting to data. Naturally, a key concept in regression analysis is to determine how well the identified equation actually models the variation in a dataset. The measure of the spread of points around the regression line can be presented using the coefficient of determination, R^2 , where

$$0 < R^2 \leq 1.$$

In other words, up to 100% of the dataset variation is explained by the chosen model. Calculations framework for assessing R^2 can be found in [21] [22]. While there are critics of R^2 , it is widely considered to be a convenient goodness-of-fit statistic [23].

V. RESULTS

Analyzed data indicates that ten of the 32 factors evaluated in the study are determinants of software project costs at the Swedish bank. The one-way ANOVA assessment indicates that six factors have significant impact on project costs, given a boundary of $p < 0.05$ which is recommended by [21]. Also, four of the factors that could provide regression models for describing project costs show reasonably high fit of data ($R^2 \geq 0.4$), and can thus be used to describe the data.

A. Factors having an impact on software development cost

Using one-way ANOVA, the following six factors were identified to have an impact on the software development project costs at the Swedish bank:

1) Risk classification ($p = 0.00016$)

A comparison of group means displayed a strong correlation between project cost and the assessed project risk; the costs were roughly four times greater for high risk projects than low risk projects, cf. Fig. 1.

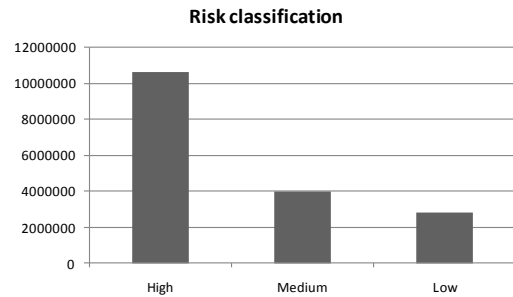


Figure 1. Project costs illustrated using *Risk classification*

2) Budget revisions ($p = 0.0033$)

Projects with many budget revisions seem to end up as more expensive than projects with few revisions (cf. Fig. 2).

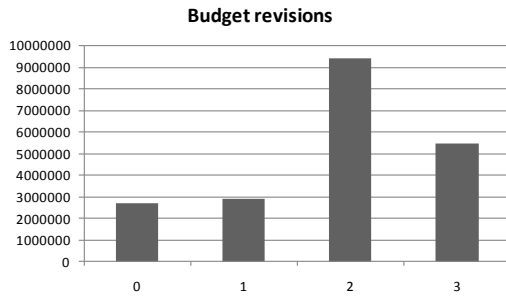


Figure 2. Project costs illustrated using *Budget revisions*

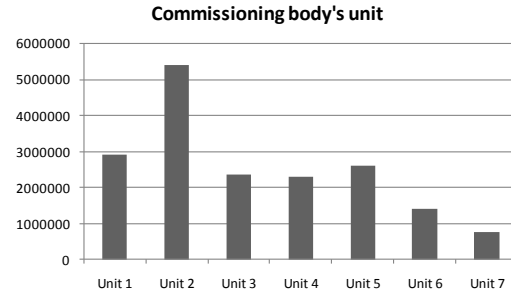


Figure 5. Project costs illustrated using *Commissioning body's unit*

3) Primary platform ($p = 0.015$)

The type software platform which is affected by a project proves to have an impact on costs, with the TDE-platform seeming to involve the most expensive projects (cf. Fig. 3).

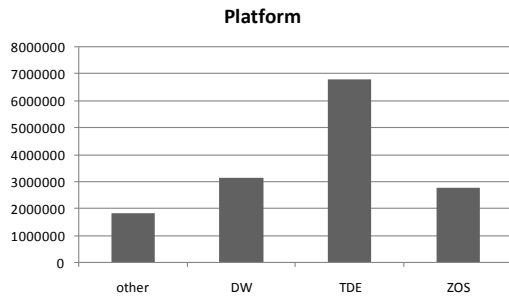


Figure 3. Project costs illustrated using *Platform*

4) Project priority ($p = 0.018$)

A project which had a high priority received approximately three times more resources than a project with low priority (cf. Fig. 4).

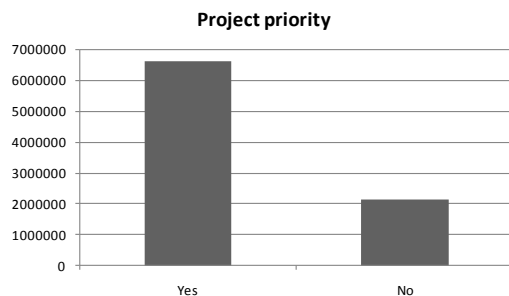


Figure 4. Project costs illustrated using *Project priority*

5) Commissioning body's unit ($p = 0.048$)

Involved project costs varied a lot with the different commissioning body units (cf. Fig. 5).

6) Commissioning body ($p = 0.052$)

The commissioning body has a high correlation to project costs (cf. Fig. 6).

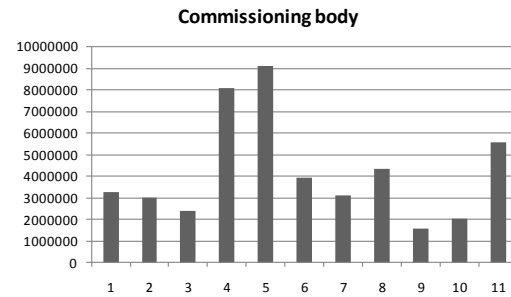


Figure 6. Project costs illustrated using *Commissioning body*

It can be argued that the sixth factor, 'Commissioning body', should have been excluded since it did not reject H_0 using the chosen boundary, however, the fact that the difference was marginal (0,2% probability) provided a reason for inclusion.

B. Modeled impact on cost

The bivariate regression analysis showed that four factors can be used to model project costs:

1) Project participants

$$(R^2 = 0.52, Y = 316366 + 311375X)$$

The number of project participants provided a reasonably good linear model describing project costs. As shown in Fig. 7, every added project participant resulted in an added cost of approximately 300 000 SEK.

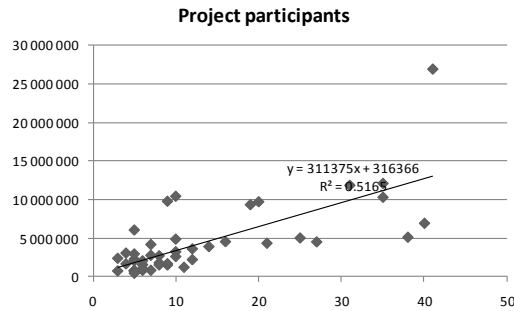


Figure 7. Project costs modeled using *Project participants*

2) Duration

($R^2 = 0.48$, $Y = 26992X - 3000000$, where $Y \geq 0 \forall X$)
Correlation between project costs and number of workdays was good enough to support a model, every added workday provided about 27 000 SEK (cf. Fig. 8).

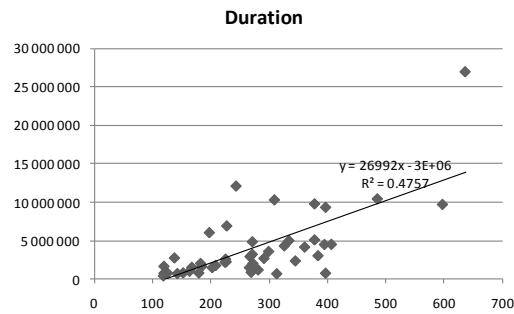


Figure 8. Project costs modeled using *Duration*

3) Function points

($R^2 = 0.46$, $Y = 977861 + 11053X$)
The number of function points showed a reasonable correlation with project cost, with approximately 11 000 SEK for every added function point (cf. Fig. 9).

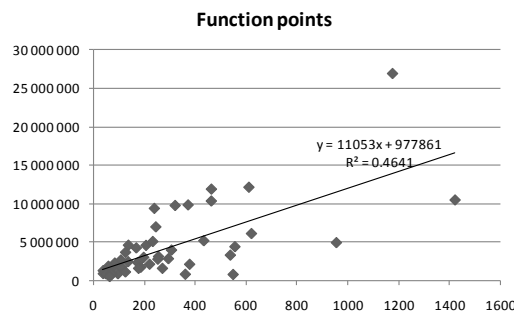


Figure 9. Project costs modeled using *Function points*

4) Consultants

$$(R^2 = 0.40, Y = 2000000 + 1000000X)$$

As shown in Fig. 10, every additional consultant increased project costs with about 1 000 000 SEK and displayed a sufficient amount of correlation.

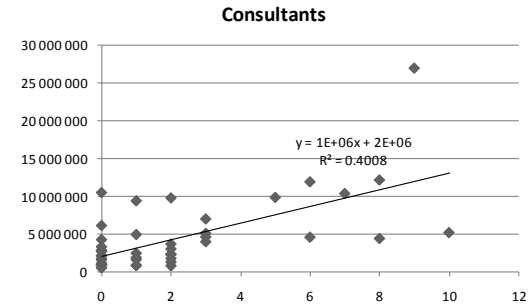


Figure 10. Project costs modeled using *Consultants*

C. Non-significant factors

Factors that did not show any significant correlation with project costs at the Swedish bank are presented in Table I and Table II.

Table I
INSIGNIFICANT FACTORS (ANOVA)

Factor	<i>p</i>
Cooperation	0.17
Architect	0.21
Final deadline revisions	0.22
Secondary platform	0.22
Liability for delivery	0.25
Competence performing assignment	0.26
Existence of testing conductor	0.28
Existence of overall schedule	0.29
Existence of overall testing plan	0.29
Performance of estimation- and prognosis efforts	0.30
Presentation interface	0.31
Integrations testing	0.34
Project manager	0.35
Quality of delivery	0.43
Conformance to requirements	0.50
External parts	0.62
Implementation efficiency	0.66
Business manager	0.76
Method for debet	0.83
Area of delivery	0.86
Project type	0.91

Table II
INSIGNIFICANT FACTORS (REGRESSION ANALYSIS)

Factor	R^2	Regression model
Length of pre-study	0.024	$Y = 5000000 - 7000000X$
Cost of pre-study	0.023	$Y = 5000000 - 7000000X$

VI. DISCUSSION

A. Significant factor results

Some significant factors have quite obvious logical associations with project cost, the scientific contribution from these results can thus be discussed. Factors included are duration, consultants and project participants.

More interesting is the probable impact on cost from involved risk, primary software platform, function points, commissioning body and commissioning body's unit.

Collected empirical data indicate that high risk projects need significantly larger budgets than low risk projects. The very nature of high risk projects imply that they contain difficult parts. Nevertheless, this study indicates that awareness of the risk status of a project could be of importance while estimating the project cost. This is a factor that none of the models in section II covers, at least not according to the terminology of [3]. The importance of careful risk handling has however been discussed in other studies, such as [24] and [25].

Regarding the factor primary software platform, if the software system developed at the Swedish bank is based on the so called TDE-platform the expenses are deemed to be relatively steeper than the other possible platforms. Computer and program attributes similar to this are vital parts of available estimation models such as COCOMO II, SEER-SEM and TEAMATe [3], [9]. This study validates the importance of including this factor.

The relation between number of function points and cost have been discussed before, see for instance [6]. Also, the function point measure is a vital part of both COCOMO II and SEER-SEM [3]. This study points in the same direction. More expensive projects due to increased system functionality is a rational, but no less interesting, conclusion.

The Standish Group states the ownership of a project as one of ten project success criteria. This study shows that the commissioning body and the commissioning body's unit significantly affect the project cost, which is in line with the Standish Group's conclusions. A reason for this might be that different commissioning bodies emphasize the requirement specification (another success factor stated by the Standish Group) to various extent [26].

The significant correlation between cost and number of revisions of project budget can be argued; the increased expenses are with little doubt implicated by other factors. To ratiocinate with certainty about this, further analysis of the underlying causes is required, see section VI-C. However, one possible explanation could be that a budget in need of several big changes was poorly performed from the beginning, which might imply shortcomings in the project plan. If this is the case, a way of measuring the quality of the project plan could be of use while estimating the software development cost, though such measurements are often difficult to perform. Nonetheless, this is an aspect of

which all models resumed by [3] are unaware of.

B. Non significant factor results

Equally interesting as the significant factor results is the circumstance that some factors often assumed to affect the software development cost failed to show a significant impact on the same in the study at the Swedish bank. Among these are:

- **Implementation efficiency** ($p = 0.66$)
- **Conformance to requirements** ($p = 0.50$)
- **Existence of overall schedule** ($p = 0.29$)
- **Project manager** ($p = 0.35$)
- **Cooperation** ($p = 0.17$)
- **Cost of pre-study** ($R^2 = 0.023$)

Since the implementation phase of a project typically require large amounts of resources [25], efficient effort during this phase was thought to provide less expensive projects. This study shows signs of this hypothesis to be true, though the difference was too small to provide a high correlation coefficient. A reason behind this could be the nature of the data collection; data was stratified according to the subjective preferences of the commissioning body connected to the project at hand.

An integral part of successful projects are to meet the specified requirements [26]. The performed case study did not assess any cost differences accompanied by conformance to requirements. However, since projects were not evaluated according to success this is hard to measure. It is also possible that the conformance of requirements correlates with the commissioning body (often responsible for the requirement specification), which did turn out to have a significant effect on development cost, see section VI-A. This could be further investigated in a multiple factorial experiment.

The existence of an overall schedule is often thought of as a key to keeping costs low [3]. The study of the Swedish bank points towards the opposite. A reason behind this could be that schedules used in the projects in hand were of poor quality, however further research is needed in order to draw any such conclusions.

A good project manager can be the difference between success and failure [27], and often determines the expenses of a project [3]. The case study of the Swedish bank did however not find that different project managers affected the project costs. This could for instance be due to the fact that all project managers at the bank are equally competent or all lacking experience to the same amount.

In [3], cooperation is included as one of the key factors affecting software development costs. The study presented in this paper could not assess any significant correlation between cooperation and costs, nevertheless, the significance level was still fairly low and the factor could thus very well turn out as significant during a follow up study. Another reason for the lack of correlation is that cooperation was

subjectively graded by each assigned commissioning body. A third prospect is that all projects might have had equally functional levels of cooperation.

A common expectation is that large projects have more expensive pre-study phases than small projects [25] and that a well performed pre-study is a good investment of time and money [26]. In line with the research performed by The Standish Group [2], [26], [27], this study indicates that it is the quality of the pre-study rather than the length or cost of the pre-study that has an effect on the project cost, where a well performed pre-study reduces the project cost. This confirms the reasoning in section VI-A, that a way of taking the quality of the pre-study into consideration could increase the accuracy of established development cost estimation models.

C. Validity and reliability

A potential problem with the external validity and reliability includes that most evaluated factors were not chosen from theory, and thus the significance of these might only be evident in the data collected during this particular case study at the Swedish bank. It is however interesting that so many factors not proven by theory excavated correlation with project costs.

Some factors had states which were graded subjectively by each commissioning body, this could affect the internal validity and reliability of the results. Another possible problem is that the factors were evaluated individually when there in fact could be correlation between factors which provide the true significant impacts on cost. Further, increased value of a factor state variable affects the states of other variables. Thus, increased cost are in no doubt the result of the combination of many changes of factor states and not just one, as is modeled in this study.

D. Industry implications

The bank in this case study documents 32 factors, which are used inter alia to forecast the software development cost. Only a few of the listed factors, e.g. number of function points, are supported by theory, see further section II. In some cases, the explanation could be that the factor of interest is difficult to measure and therefore exchanged to something more easily quantified. For instance, instead of measuring the quality of the project pre-studies, the bank documents the length and cost of the pre-studies. However, the above reasoning cannot explain the absence of all factors used in established cost estimation models. Among others, personnel attributes stated by [3] such as continuity of project team members, would not be that hard to measure.

Various established cost estimation models have been tested on several projects with good results. The concerned bank (and probably other companies as well) could most likely improve the accuracy of their cost estimations by looking more carefully at available theory on the subject.

VII. CONCLUSIONS

Software systems of today are often complex, making development costs difficult to estimate. This paper uses data from 50 projects performed at one of the largest banks in Sweden to identify factors that have an impact on software development cost. In total 32 factors were studied. Correlation analysis of the relationship between factor states and project costs were assessed using one-way ANOVA and bivariate regression analysis.

This study validates the use of factors such as function points and primary software platform, which can be found in established software development cost estimation models, such as COCOMO II and SEER-SEM. The risk classification of a project was another factor found to affect the project cost. Accordingly, this study indicates that including information about the risk classification would increase the accuracy of established models, as they so far do not consider this explicitly.

A somewhat more speculative suggestion is to consider the quality of the pre-study in the cost estimation. This however requires further research regarding the intergroup dependences.

Two factors often assumed to affect the project cost are the efficiency of the implementation and the cost of the pre-study. In this study though, these factors failed to display a significant correlation with project cost.

A. Future work

To reinforce the results of this study and further investigate the underlying causes of the results, multiple factorial experiments of the data could be performed. Also, further research on the suggested extensions of the established cost estimation models would be a useful validation of this study.

REFERENCES

- [1] L. Laird and C. Brennan, *Software Measurement and Estimation: A Practical Approach*. IEEE Computer Society / John Wiley & Sons, 2006.
- [2] The Standish Group, "Chaos summary 2009," Available on <http://www.standishgroup.com>, accessed February 5, 2010, 2009.
- [3] B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approaches – a survey," *Annals of Software Engineering*, vol. 10, pp. 177–205, 2000.
- [4] A. Albrect and J. Gaffney, "Software function, source lines of code, and development effort prediction: A software science validation," *IEEE Transactions on Software Engineering*, vol. SE-9, no. 6, November 1983.
- [5] The Common Software Measurement International Consortium (COSMIC), "The COSMIC Functional Size Measurement Method, Measurement Manual," COSMIC, International standard version 3.0.1, 2009.

- [6] J. E. Matson, B. E. Barrett, and J. M. Mellichamp, "Software development cost estimation using function points," *IEEE Transactions on Software Engineering*, vol. 20, no. 4, 1994.
- [7] B. Boehm, *Software Engineering Economics*. Prentice Hall, 1981.
- [8] S. Devnani-Chulani, "Bayesian analyses of software cost and quality models," Ph.D. dissertation, University of Southern California, 1999.
- [9] R. Lagerström, P. Johnson, D. Höök, and J. König, "Software change project cost estimation - a bayesian network and a method for expert elicitation," in *Third International Workshop on Software Quality and Maintainability (SQM 2009)*, Mars 2009.
- [10] R. Lagerström, U. Franke, P. Johnson, and J. Ullberg, "A method for creating enterprise architecture metamodels - applied to systems modifiability analysis," in *the International Journal of Computer Science and Applications*, vol. 6, no. 5, 2009, pp. 89–120.
- [11] Galorath Inc., "System Evaluation and Estimation of Resources - Software Estimation Model (SEER-SEM)," Available on <http://www.gaseer.com>, accessed February 5, 2010.
- [12] R. Park, "The central questions of the price software cost model," in *4th COCOMO Users Group Meeting*, November 1988.
- [13] H. A. Rubin, "Macroestimation of software development parameters: the estimacs system," in *SOFTFAIR Conference Development Tools, Techniques and Alternatives*. IEEE Press, July 1983, pp. 4–16.
- [14] C. Jones, *Applied Software Measurement*. McGraw Hill, 1997.
- [15] R. C. Tausworthe, "Deep space network software cost estimation model," in *Jet Propulsion Laboratory Publication 81-7*, 1981.
- [16] L. Putnam and W. Myers, "Measures for excellence," *Yordon press Computing Series*, 1992.
- [17] R. W. Jensen, "A comparison of the Jensen and COCOMO schedule and cost estimation models," in *Proceedings of the International Society of Parametric Analysts*, April 1983, pp. 96–106.
- [18] J. Bailey and V. R. Basili, "A meta-model for software development resources expenditures," in *5th International Conference on Software Engineering*. IEEE Press, 1981, pp. 107–116.
- [19] 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, January 2007.
- [20] K. Molokken-Ostfold and N. C. Haugen, "Combining estimates with planning poker an empirical study," April 2007.
- [21] R. Warner, *Applied Statistics*. SAGE Publications, 2008.
- [22] D. Montgomery, *Design and Analysis of Experiments*. John Wiley and Sons, 2009.
- [23] G. King, "How not to lie with statistics: Avoiding common mistakes in quantitative political science," *American Journal of Political Science*, vol. 30, No. 3, pp. 666–687, 1986.
- [24] P. M. Institute, *Project Management Body of Knowledge*, 4th ed. Crosswind Project Management Inc., 2009.
- [25] P. W. G. Morris, *The management of projects*. Thomas Telford, 1994.
- [26] The Standish Group, "Chaos Report," The Standish Group International, Inc., Tech. Rep., 1995.
- [27] —, "What are your requirements?" The Standish Group International, Inc., Tech. Rep., 2003.