COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment

Lionel C. Briand, Khaled El Emam, Frank Bomarius
Fraunhofer Institute for Experimental Software Engineering (IESE)
Sauerwiesen 6
D-67661 Kaiserslautern
Germany
{briand,elemam,bomarius}@iese.fhg.de

International Software Engineering Research Network Technical Report ISERN-97-24
(Revision 2)

COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment

Lionel C. Briand, Khaled El Emam, Frank Bomarius
Fraunhofer IESE
Sauerwiesen 6
D-67661 Kaiserslautern
Germany
{briand,elemam,bomarius}@iese.fhg.de

Abstract

Current cost estimation techniques have a number of drawbacks. For example, developing algorithmic models requires extensive past project data. Also, off-the-shelf models have been found to be difficult to calibrate but inaccurate without calibration. Informal approaches based on experienced estimators depend on estimators' availability and are not easily repeatable, as well as not being much more accurate than algorithmic techniques. In this paper we present a method for cost estimation that combines aspects of algorithmic and experiential approaches (referred to as COBRA, COst estimation, Benchmarking, and Risk Assessment). We find through a case study that cost estimates using COBRA show an average ARE of 0.09, and show that the results are easily usable for benchmarking and risk assessment purposes.

1 Introduction

Project and program managers require accurate and reliable cost estimates to allocate and control project resources, and to make realistic bids on external contracts. They also need to determine whether, for a given system size, a budget is realistic, the cost of a prospective project is likely to be too high (risk assessment), or whether a project is of comparable difficulty with previous, typical projects in an organization (benchmarking). Such analyses may lead to the redefinition of the project's requirements or to the definition of appropriate contingency plans.

Different techniques for cost estimation have been discussed in the literature [3][9][10], for example: algorithmic and parametric models, expert judgment, formal and informal reasoning by analogy, price-to-win, top-down, bottom-up, rules of thumb, and available capacity. More recently, analogical and machine learning models [5][23] have also been developed.

Despite extensive development of algorithmic models over the last twenty years, recent surveys indicate that very few organizations actually use them. For instance, a survey of 364 Dutch companies that estimate costs found that less than 14% used models [9] and instead produce their cost estimates largely on experiential based approaches, such as expert opinion or by examining documentation from previous projects. One survey of software development within JPL found that only 7% of estimators use algorithmic models as a primary approach for estimation [10]. Only 17% of respondents in another survey used off-the-shelf cost estimation software, which usually embody some form of algorithmic estimation model [17]. In fact, the most extensively used estimation approach was found to be "comparison to similar, past projects based on personal memory".

There are a number of possible reasons why algorithmic and parametric models are not used extensively. These include the fact that many organizations do not collect sufficient data to allow the construction of such models. For example, one survey found that 50% organizations do not collect data on their projects [9]. Another survey reported that for 300 measurement programs started since 1980, less than 75 were considered successful in 1990, indicating a high mortality rate for measurement programs [20]. Another reason is that many of these models are not very accurate. For example, one reported study demonstrated that model-based estimates do not perform considerably better than experiential-based estimates [16]. Another survey found that there is essentially no difference in the percentage of projects that overrun their estimates between users and non-user of off-the-shelf cost estimation tools [17]. Furthermore, an evaluation of different models showed gross overestimation [13][16], and comparisons of estimates produced by different models for the same project exhibited wide variation [13][19][18][16].

Experiential approaches have their own drawbacks as well. First of all, increaing use of some informal estimation approaches, such as guessing and intuition, have been found to be related to increases in the number of large projects that overrun their estimates [17]. In addition, more structured bottom-up experiential approaches can potentially suffer from an optimistic bias due to estimators extrapolating from an estimate of only a portion of the system [3][9]. In one study that evaluated experiential estimates, optimism was demonstrated [10]. However, another study showed that experiential estimates are pessimistic, with the estimators estimating more than the actual [16]. In any case, both over and underestimation, as noted in [17], can have negative ramifications on a project and its staff. Furthermore, it has been shown in one study that the accuracy and variation of experiential-based estimates is affected significantly by application and estimation experience of the estimators, with more experienced estimators performing better [10]. However, in a given organization, it will seldom be possible to find available, highly experienced estimators for every new project. Given the prevalent reliance on informal approaches to cost estimation, many of these estimates would not be easily repeatable either even if the most experienced estimators were available.

The disadvantages of current cost estimation approaches have prompted some to urge the use of multiple estimation approaches together [9][3]. This is intended to alleviate the disadvantages of individual approaches.

It has been reported that risk analysis features are not satisfactory on many algorithmic models and their tool implementations [9]. However, recently techniques for

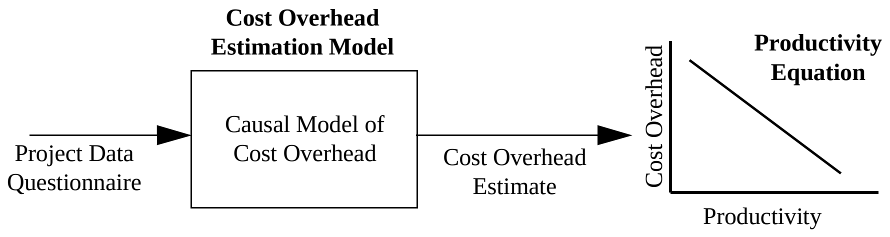
Cost Overhead Estimation Model
Causal Model of Cost Overhead
Project Data Questionnaire
Cost Overhead Estimate
Productivity Equation
Cost Overhead
Productivity

Figure 1: Overview of productivity estimation model.

taking into account the probability of cost drivers having certain values in conjunction with the COCOMO model [7], and a tool that allows adjusting effort estimates based on the probability of certain events occurring have been developed [14]. Neither of these approaches allow the project manager to determine whether the cost of a project is too high.

For the more structured experiential approaches, potential risks can be accounted for when producing a cost estimate [8]. However, this would not indicate to the project manager whether the cost of the project was too high.

In this paper we report on a hybrid cost modeling method, COBRA: COst estimation, Benchmarking, and Risk Assessment, based both on expert knowledge (i.e., experienced project managers) but also on quantitative project data. However, it does not require extensive past project data bases to be operational. It is repeatable and follows a fully explicit rationale. In order to illustrate the method and demonstrate its feasibility, we provide the results of a case study where a local cost model and procedures for risk assessment and benchmarking were developed.

Section 2 presents the principles of our method. In Section 3 we describe the steps of the method and the underlying model in detail. In Section 4, the validation of our method on a case study is given. Section 5 describes how to perform benchmarking and risk assessment using the model presented in previous sections. We conclude the paper in Section 6 with a summary and directions for future work.

2 Principles of the Method

The core of our method is to develop a productivity estimation model. The productivity estimation model has two components as illustrated in Figure 1. The first component is a model that produces a *cost overhead estimate*. The second component is an productivity model that estimates productivity from this cost overhead[1]. Because the first component is independent of any used project size measure, it may feed several productivity models, based on different size measurement. This allows the use of a unique cost overhead model despite different types of project requiring different size measurements and despite changes in the corporate size measurement program. Since, as discussed below, the productivity model requires less effort to develop and maintain, this is a very strong economic argument for the use of COBRA.

1.It is important to note that the issues related to the sizing of projects will not be addressed in this paper and that our method is independent of the sizing measure that is used. The purpose of the size measure used in this paper's case study is only to exemplify the model usage and demonstrate its validity.

Figure 1 clearly illustrates the hybrid nature of our approach. The cost overhead estimation model is developed based on project managers' experience and knowledge. The productivity equation is developed using past project data.

Below we explain the basic principles of the model and the equations, and how they can be used for cost estimation, benchmarking, and risk assessment.

2.1 Cost Overhead

The cost overhead estimation model takes as input a set of data that characterizes a particular project. This data is collected via the *project data questionnaire*.

The model predicts the *cost overhead* resulting from suboptimal conditions associated with a project. The cost overhead is expressed as an additional percentage on top of the cost of a project run under optimal conditions. This is referred to as a *nominal* project. A nominal project is a hypothetical ideal (e.g., the objectives are well defined and understood by the project staff and the client, and all of the key people on the project already have the right capabilities for the project). Real projects will almost always deviate significantly from the nominal project. Cost overhead is intended to capture the extent of this deviation. The reason we used the notion of cost overhead is that experts (i.e., experienced project managers) could easily relate to it and predict the effect of cost drivers in terms of such a percentage, independently of project size.

2.2 Productivity Model

A nominal project has the highest possible productivity. Real projects that deviate from the nominal will have lower productivity. We expect (and demonstrate during the validation) that the cost overhead estimate is strongly related to productivity. Specifically, the relationship is:

$$P = \beta_0 - (\beta_1 \times CO) \qquad \text{(eqn. 1)}$$

where P is productivity, and CO is the cost overhead. This is the productivity equation in Figure 1.

The cost overhead estimation model is defined such that this relationship is linear. The β_0 parameter is the productivity of a nominal project. The β_1 parameter captures the slope between CO and P. In practice, the two beta parameters above can be determined from a small historical data set of projects (~10) since only a bivariate relationship is modeled.

Although values of CO can theoretically yield null or negative P values, this is not a problem in practice. First, when such a productivity model is built, it should only be used for interpolation purposes and not outside the range of values upon which it has been built. Second, if such a high

CO value is actually encountered when trying to estimate productivity, this probably means either that the project is undoable under similar conditions or that it is of very different nature than that of the projects on which the productivity model has been built.

There is no specific reason to model the relationship between CO and P as linear. Any relationship yielding a better fit should be considered. However, when one has only a small number of project data points available, this is likely to be a reasonable compromise.

2.3 Estimating Cost

It has been suggested that software projects exhibit economies or diseconomies of scale [2]. However, a recent empirical analysis provided evidence that does not support this contention, and concludes that the relationship between effort and size is linear [15]. We therefore assume that the relationship between effort and size is linear. This can be expressed as:

$$\text{Effort} = \alpha \times \text{Size} \qquad \text{(eqn. 2)}$$

In eqn. 2, both the α value and the size are variables that are determined for each project. The α value is given by:

$$\alpha = \frac{1}{\beta_0 - (\beta_1 \times \text{CO})} \qquad \text{(eqn. 3)}$$

The effort for a particular project can be determined from eqn. 2 using the system size and cost overhead.

As one deviates away from the nominal project the cost overhead increases and therefore the α value increases, indicating that for the same system size, more effort is expended (i.e., lower productivity).

2.4 Project Cost Risk Assessment

The concept of cost risk that we employ is the probability, for a given project, to overrun its budget, or any additional percentage above it. In the model presented above, regardless of the strategy used, the computation of a budget B translates into α_B and CO_B values. The first step is to compute the probability of having an CO/α value greater than CO_B/α_B. Then the maximum tolerable probability P_M that a given project has less productivity ($1/\alpha_B$) than that required by the budget has to be defined. Decisions regarding risk assessment can then be made as follows: If

$$P(\alpha \geq \alpha_B) > P_M \qquad \text{(eqn. 4)}$$

is satisfied then the project would be considered as high risk and preventative actions as well as contingency plans would be necessary. This approach can be extended to multiple thresholds corresponding to different budget overheads. In such a case, multiple levels of risk can be defined and different action plans associated with each risk level. The implementation of this approach will be further detailed in Section 5.

2.5 Project Cost Benchmarking

Cost benchmarking follows similar principles. Their main difference resides in the way CO/α threshold values are defined. In our context, benchmarking involves comparing the CO of a given new project with a historical database of representative projects. The goal is to assess whether this project is likely to be significantly more difficult to run than the “typical” project in the organization and include more cost overhead. Such a comparison may lead to decisions regarding the staffing of the project or the contractual agreement with the customer. Similarly to risk assessment, we may define thresholds which correspond to the CO median (CO_T for “typical”) or upper quartile (CO_M for “majority”) in the organization’s project database. Then, the probability of lying above such thresholds is computed and used to decide of the likely relative difficulty level of the new project: above typical, above majority. This is further described in Section 5.

2.6 Estimating Cost Overhead

Central to our whole approach is being able to come up with a cost overhead estimate that meets the assumptions of eqn. 1 for any project. We do this through a *causal model* of factors affecting the cost of projects within the local environment under study.

A causal model consists of *cost factors* or *drivers* and *relationships* with cost overhead or amongst the factors themselves. The relationships specify the nature of the impact on cost overhead. There are two types of relationships: *direct* and *interaction*. A direct relationship means that the factor directly increases or decreases cost overhead irrespective of the values of the other factors. An interaction relationship means that the manner in which a factor affects cost overhead depends on the value(s) of one or more other factors. In the causal model that we develop we limit the number of factors in an interaction to a maximum of three.

In order to simplify the modeling process, it is important that all the factors that are related to cost are modeled in such a way so that they can be considered additive. This means that each factor increases/decreases cost by a certain amount, and that the overall impact on cost is the sum of the affect of each of the individual factors. To the extent possible, the non-additive properties of the model should all be captured through interactions.

2.7 Validation of the Method

To validate our approach, it is necessary to empirically demonstrate two things. First, that the assumption underlying eqn. 1 is valid (i.e., that cost overhead is strongly related with productivity). Second, that eqn. 2 provides accurate estimates of effort. Using our case study, we present an initial validation of our method.

3 Estimating Cost Overhead

In this section we present the steps of the method to develop a cost overhead estimation model in detail. Each step is presented in terms of its objectives, the inputs, the process that is followed, and the outputs. We also illustrate each step with reference to a case study where we applied our method. This makes it easier to understand the method’s steps.

The case study we present here took place in a software house, software design & management (sd&m), involved mainly in the development of MIS software in Germany. The development of the model took a total effort of three man-months, including the writing of reports and the development of a prototype. We were able to collect project questionnaire data on 9 projects and reliable effort/size data on 6 of them. This was, of course, not enough to develop a data driven model but was good enough for an initial hybrid model developed using COBRA.

3.1 Identify the Most Important Cost Drivers

Objectives

The cost of software projects is driven by many factors. While one could include all of the possible cost drivers that have been presented in the literature when developing a cost model, it has been argued that only a subset of factors are relevant in a particular environment. Therefore, we identify the cost factors that are most relevant for the environment under study.

Process

The literature on software engineering cost estimation and productivity was reviewed to identify potential cost drivers. Some of the articles that were of most influence were [4][28][1][21]. Based on this review and after removal of redundancy, an initial list of *Product*, *Process*, *Project*, and *Personnel* category cost drivers was drafted.

Eleven experienced project managers were then asked to go through the cost drivers and to comment on their clarity and ease of understanding during an interview with the authors. This is to ensure that different project managers interpret the cost drivers in the same way. Furthermore, they were requested to comment on the completeness of the cost drivers in each category (i.e., are there any missing), their relevance (i.e., should this cost driver be considered at all), whether they ought to be further refined, and on any overlaps (i.e., cost drivers that ought to be combined). The cost driver definitions were subsequently revised, leaving a total of 39 cost drivers.

Then, the cost drivers were ranked according to the magnitude of their impact on cost. The eleven project managers were requested to rank the cost drivers within each category. They were presented with slips of paper with the cost driver written on it (presented in random order) and asked to order them according to their impact on cost. Ties are allowed. The average of the raw ranks was used to arrive at the final ranking of the cost drivers within each category.

During the analysis of raw ranks, it is important to look at the variation in ranking and also at the relationship between the deviation from the average and the experience of the respondent. Both of these types of analysis allow us to determine how much consensus there is amongst the project managers and to identify outliers in responses. Since not all experts can be expected to be knowledgeable about all factors, outlier analysis is used to identify responses to be further investigated and possibly removed. This reduces considerably the ranking variance.

The ranks within each category are then used to eliminate the least important cost drivers from further consideration. The number of factors retained is decided according to the resources available to develop the cost risk model. We opted to retain a total of twelve factors.

Outputs

The output of this step is a minimal set of cost drivers that have the largest impact on the cost of projects in the local environment.

Case Study

We only present the detailed results for the Product category here. The average ranks are shown in Figure 2. These are based on data collected from 11 project managers. The first data column gives the average ranks for each cost driver. The standard error of the rank is computed as the standard deviation of the ranks. The standard error gives an indication of distance from the "true" ranking of a cost driver, assuming that the project manager population's average rank is an unbiased estimate of this "true" ranking.

As can be seen, the importance of software reliability, software usability, and software efficiency were considered to have the largest impact on cost within this environment. The least important was data base size and the complexity of computational operations. For the former, it was believed that existing tools can manage their data base sizes adequately. For the latter, the existence of complex computational operations was believed to be sufficiently rare in this environment that this factor would not vary.

A more general measure of the extent of agreement amongst the ranking provided by the respondents is Kendall's coefficient of concordance [22]. For the product cost driver, this was 0.38 and significant at an alpha level of 0.1, thus showing a significant level of concordance between all respondents.[1]

When we considered the relationship between the "error" (computed as difference from the average rank) and experience, a negative relationship was witnessed (Spearman's correlation of -0.48, significant at an alpha level of 0.1). This indicates that as the number of years of project management experience increases (and by implication, estimating experience), the error decreases. Therefore, we removed the least experienced respondents from our data set and recomputed the average rank and standard error (see Figure 2). As can be seen the standard error for many of the cost drivers tended to decrease. Kendall's coefficient of concordance increased to 0.54 (significant at an alpha level of 0.1), hence indicating greater agreement amongst the respondents.

Based on this final ranking, we retained the top ranking twelve cost drivers. For the Product category, we retained the top three ranked factors. The retained Personnel factors were: Consistency of stakeholder objectives and culture, ability and willingness of stakeholders to accommodate other stakeholders' objectives, and analyst (or key staff) capability. The retained Process factors were: extent of customer participation and extent of disciplined requirements management. The retained Project factors were: requirements volatility, extent to which the stakeholders understand the product objectives, need for innovative data processing architectures and algorithms, and development schedule constraints.

3.2 Develop Qualitative Causal Model

Objectives

The manner in which cost drivers have an impact on the cost of projects can be complex. In particular, the cost drivers may interact with each other and this interaction influences the cost. The purpose of this step is therefore to capture the

1. We use an alpha level of 0.1 here so as to maintain reasonable statistical power levels considering the low number of respondents for such an analysis.

		All Respondents		Most Experienced Respondents Only	
		Avg. Rank	Std. Err.	Avg. Rank	Std. Err.
PROD.1	Importance of software reliability	4.54	3.64	3.62	3.25
PROD.2	Importance of software usability	4.81	2.86	4.12	1.96
PROD.3	Data base size	8.18	3.22	8.00	2.56
PROD.4	Importance of software efficiency	5.18	3.25	4.12	2.36
PROD.5	Documentation match to life-cycle needs	7.27	4.15	removed[a]	removed[a.]
PROD.6	Importance of software maintainability	5.72	4.17	5.25	3.81
PROD.7	Importance of software portability	7.36	3.61	6.00	3.25
PROD.8	Importance of software reusability	6.72	4.20	6.12	3.56
PROD.9	The complexity of the control operations	6.54	3.36	6.12	3.09
PROD.10	The complexity of the computational operations	7.45	4.06	7.87	3.80
PROD.11	The complexity of device dependent operations	8	3.90	7.62	3.50
PROD.12	The complexity of user interface management operation	4.64	3.69	5.00	3.70
PROD.13	The complexity of data management operations	6.36	3.64	7.00	3.50

a. According to the comments we obtained during the interviews, the factor PROD.5 was not adequately defined for this organization's context. We therefore removed PROD.5 from the list of factors.

Figure 2: Rankings of the *Product* cost drivers.

manner in which the cost drivers affect cost explicitly.

Inputs
The minimal set of cost drivers selected in the previous step.

Process
To build a qualitative causal model, the project managers were requested to go through each of the most important cost drivers and explain why they think it affects cost, how, and if there are any other cost drivers they think are important to consider at the same time in determining its impact (i.e., interaction). This is done during an interview with each project manager separately.

Based on the comments from a number of project managers, an initial version of the qualitative causal model was developed. This is then subsequently validated with the project managers to ensure that it accurately reflects their expert opinion about how the factors have an impact on cost. It should be noted that new factors may be introduced at this stage if they interact with one of the already selected 12 factors (i.e., moderate their influence on cost).

Validation proceeds by going through each relationship and explaining the causal mechanism that is captured, including any interactions. This is done with each project manager. If the project manager agrees with the explanation, then the relationship is validated.

Outputs
A qualitative causal model.

Case Study
The validated qualitative causal model that was developed is shown in Figure 3. This model reflects the collective experiences of the organization's senior project managers about the factors that affect cost of development.

Where one lined arrow points at another, this indicates an interaction, for example, between customer participation and customer competence. In this case, increased customer competence magnifies the negative relationship between customer participation and cost.

3.3 Develop Project Data Questionnaire

Objectives
To use the cost overhead estimation model, the project managers will have to be able to characterize their projects in terms of the factors in the causal model. This is achieved through a questionnaire. The purpose of this step is to develop a reliable questionnaire for the collection of this data.

Process
Each factor in the causal model is decomposed into a number of orthogonal *variables* that measure that factor through questions. The authors performed this decomposition and validated the orthogonality assumption by having project managers review the variable definitions and associated questions.

Each variable has to be measured in the questionnaire. We assume that each variable is measured on an interval or approximately interval scale. Therefore, it is necessary to take steps to ensure that this assumption is not extensively violated. In addition, we ensured that responses to the questionnaire are reliable.

Most of the questions were of the Likert-type. This consists of a statement, followed by a number of response categories that the respondent chooses from. Three types of questions were used in the questionnaire: *frequency*, *evaluation*, and *agreement* (these are commonly used types of scales [24]). Frequency type scales ask the respondents about how many times the activity described in the statement

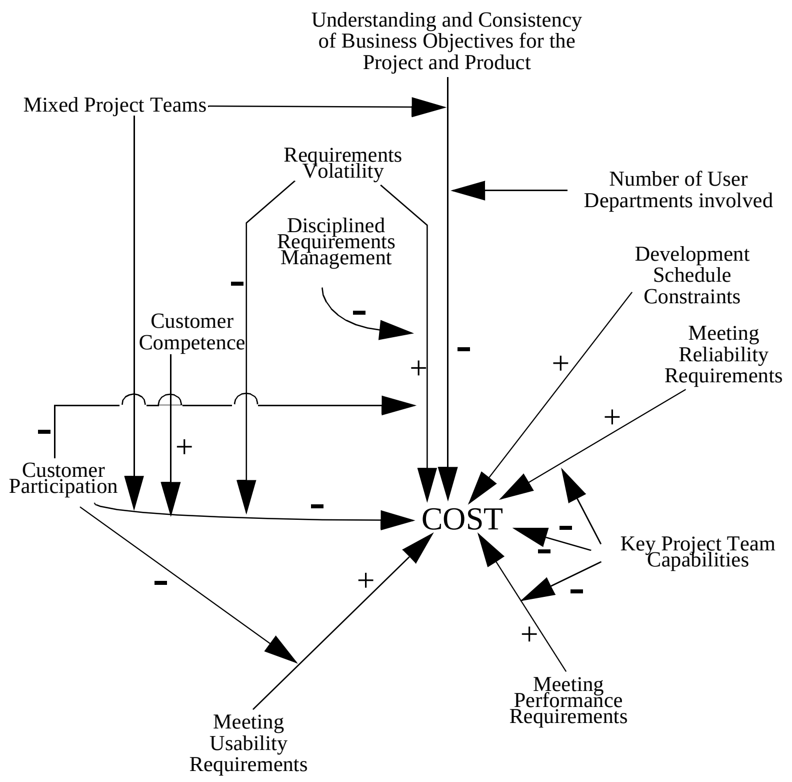
Understanding and Consistency of Business Objectives for the Project and Product
Mixed Project Teams
Requirements Volatility
Number of User Departments involved
Disciplined Requirements Management
Development Schedule Constraints
Meeting Reliability Requirements
Customer Competence
Customer Participation
COST
Key Project Team Capabilities
Meeting Performance Requirements
Meeting Usability Requirements
+
-

Figure 3: The causal model. The straight lines indicate direct relationships. The +'s indicate a positive relationship, and a -'s indicate a negative relationship.

would happen. Evaluation type scales are used to rate the capability of a key person or people on the project along a good-bad spectrum. Agreement type scales ask respondents about their extent of agreement to the statement given. The statement is usually a characterization of the project.

The response categories chosen follow the guidelines developed by Spector for equally spaced response categories [26]. During pilot testing of the questionnaire however, we found that some of the respondents had difficulty interpreting the agreement scale that used Spector's equally spaced response categories. We therefore used a very common agreement scale instead. A subsequent study by Spector [25] indicated that whether scales used have equal or unequal intervals does not actually make a practical difference. In particular, the mean of responses from using scales of the two types do not exhibit significant differences, and that the test-retest reliabilities (i.e., consistency of questionnaire responses when administered twice over a period of time) of both types of scales are both high and very similar. He contends, however, that the unequal scales are more difficult to use, but respondents conceptually adjust for this. The three scales that we used are summarized in Figure 4.

To ensure that the scales are reliable, we collected data from two senior people on each of three projects. We then compared the responses to the questions. For most questions, the responses to the questionnaire were similar or the same, irrespective of the respondent, and we concluded the reliability of the questionnaire was satisfactory. However, a few questions were modified to improve their interpretability and the consistency of responses..

Outputs

The results of this step is a validated project questionnaire for measuring each of the factors.

Case Study

The final questionnaire contains a series of questions for each factor. The complete questionnaire is presented in the appendix.

Frequency Scale

❐ Rarely
❐ Infrequently
❐ Occasionally

Evaluation Scale

❐ Inferior
❐ Unsatisfactory
❐ Satisfactory

Agreement Scale

❐ Strongly Agree
❐ Agree
❐ Disagree

Figure 4: The three types of subjective scales used.

3.4 Quantify Relationships

Objectives

Now that we have a qualitative causal model, it is necessary to quantify it. In order to quantify the model we need to determine the magnitude of each of the relationships in the qualitative causal model.

Process

We rely on expert input to quantify the relationships. The quantification is the percent cost overhead above that of a nominal project. These are called cost overhead *multipliers*. The percentage value is assigned for the extreme values on the project questionnaire. For example, if we have a relationship between knowledge of the application domain and cost overhead, we would request the percentage overhead at the extreme situations for this variable: knowledge is excellent and knowledge is inferior. However, in this particular example, excellent knowledge coincides with our nominal project, so there is no overhead above a nominal project for that situation, and the respondent only needs to provide the percentage overhead for the inferior knowledge situation. So if the respondent gave a value of 20%, that would mean that if the knowledge of the application domain is inferior, then the cost of the project would be 120% that of a nominal project. It should be noted that exactly the same terms are used to describe the extreme situations as in the questionnaire.

This multiplier information is collected from multiple project managers and then aggregated. The nature of this aggregation is explained later on.

Project managers will generally have difficulty giving precise cost overhead percentage values for different situations. Therefore, it is more appropriate to ask them for distributions, thus taking into account the uncertainty in their responses.

It is preferable to model cost multipliers as a distribution rather than as a single value for a number of reasons:

1. In reality, the effect of a single variable will vary depending on the values of other factors. We have captured the most important of these types of relationships in the form of interactions. However, we did not represent all possible interactions in our causal model, only the ones that were deemed to be most important. Therefore, each cost multiplier is expected to vary about a central value depending on the values of other factors that we do not explicitly consider in the causal model.
2. It is often easier for experts to give *minimum*, *most likely*, and *maximum* values than it is to give single values. The reason is that the 3 values capture the uncertainty of the expert about the value of the cost multiplier. The wider the distribution the more uncertain the expert is. It is to be expected that some uncertainty will exist when dealing with complex phenomenon such as the cost of software projects.

The distribution we use to model expert judgements of cost multipliers is the triangular one. In many practical situations, either a triangular or BetaPERT distribution (see [6]) is used to model expert knowledge of a variable where there are maxima, minima and a central tendency/most likely value [27]. However, it is acknowledged that the triangular distribution is adequate in general [27], and for modeling cost-related risk [8] since it is simpler and we have no rationale to adopt more complex distributions. Although the triangular distribution gives more weight to the minima and maxima when compared, for example, to a BetaPERT distribution, the experts interviewed here provided what is referred to as "practical" minima/maxima [27], that is extreme but plausible situations. Furthermore, in previous software engineering studies, a triangular distribution has been used to model subjective uncertainty in cost estimation [11][10] .

To collect this multiplier information, the data collection tables exemplified in Figure 5 were used. A respondent provides actual values in the form where there are terms in the table. To understand these terms, it is first necessary to introduce some general notation:

$$f = 1 \ldots k, \text{ is an index for each factor F}$$

Regarding the project questionnaire, each question q and question responses on the four-point measurement scale are formalized as follows:

$$Q_{f,q} = \{Q_{f,q,0}, Q_{f,q,1}, Q_{f,q,2}, Q_{f,q,3}\}$$

$$Q_{f,q,i} = \begin{cases} 1, & \text{if response category i is selected} \\ 0, & \text{if response category i is not selected} \end{cases}$$

Each question corresponds to a variable which is assigned a value as follows:

$$V_{f,q} = \{i \mid i \in [0,3], Q_{f,q,i} = 1\}$$

Let $HCO_{f,q}(V_{f,q})$ denote a function that returns the maximum multiplier value for a response of $V_{f,q}$. Also, let $MCO_{f,q}(V_{f,q})$ and $LCO_{f,q}(V_{f,q})$ denote the same for the most likely multiplier value and the minimal multiplier value respectively. Also let, $HCO_{f,q}(V_{f,q}, V_{w,r})$ and $HCO_{f,q}(V_{f,q}, V_{w,r}, V_{y,s})$ denote the same for the case of two way and three way interactions.

For example, let's consider the direct relationship in Figure 5. The f value refers to the "Key Project Team Capabilities" factor. The q value refers to the question in the questionnaire asking about platform knowledge. In the questionnaire, question q uses an evaluation scale. If the response is "Inferior" then the $V_{f,q}$ value is 3. While the multiplier form is being filled up, the respondent assigns multipliers when the $V_{f,q}$ value is 3 (i.e., extreme departure from the nominal case). If the *most likely* multiplier value is, say 20%, then the respondent is saying s/he would add 20% to the cost of a nominal project if all key persons on the project team have inferior knowledge and familiarity of the platform to be used. For the two-way interaction the respondent has to do this twice, one for each of the extreme values of the interaction variables. For the three-way interaction, the respondent has to do this four times, one for each of the combinations of the extreme values on the two interaction variables.

Direct Relationship:

There is <u>no</u> <u>person</u> on the project team with sufficient familiarity and comprehension of the platform to be used (e.g., programming language(s), Operating System, database management systems):

Multiplier (overhead above nominal project)	$HCO_{f,q}$ (3) / $LCO_{f,q}$ (3) / $MCO_{f,q}$ (3)

Two way interaction:

Interface specifications to software that is being developed in parallel are going to change

Multiplier (overhead above the nominal project) Disciplined Requirements Management	
Nominal	High (extreme)
$HCO_{f,q}$ $(0,M_{w,r})$ / $LCO_{f,q}$ $(0,M_{w,r})$ / $MCO_{f,q}$ $(0,M_{w,r})$	$HCO_{f,q}$ $(3,M_{w,r1})$ / $LCO_{f,q}$ $(3,M_{w,r})$ / $MCO_{f,q}$ $(3,M_{w,r})$

Three way interaction:

The system functionality is completely new for the customer

Multipliers (overhead above the nominal project)		User Participation	
		Low (extreme)	Nominal
Disciplined Requirements Management	Nominal (follow organizational standards)	$HCO_{f,q}$ $(3,M_{w,r},M_{y,s})$ / $LCO_{f,q}$ $(3,M_{w,r},M_{y,s})$ / $MCO_{f,q}$ $(3,M_{w,r},M_{y,s})$	$HCO_{f,q}$ $(3,M_{w,r},0)$ / $LCO_{f,q}$ $(3,M_{w,r},0)$ / $MCO_{f,q}$ $(3,M_{w,r},0)$
	High (extreme)	$HCO_{f,q}$ $(3,0,M_{y,s})$ / $LCO_{f,q}$ $(3,0,M_{y,s})$ / $MCO_{f,q}$ $(3,0,M_{y,s})$	$HCO_{f,q}$ $(3,0,0)$ / $LCO_{f,q}$ $(3,0,0)$ / $MCO_{f,q}$ $(3,0,0)$

Figure 5: Examples of forms used to collect the multiplier data.

Outputs

The output from this step is a set of multipliers, reflecting the expert opinion and its inherent uncertainty, for each of the relationships in our causal model.

Case Study

In our case study we obtained multipliers from 7 different project managers for each of the relationships.

3.5 Operationalize Cost Overhead Estimation Model

Objectives

To produce a cost overhead estimate, the multipliers and the project questionnaire variables have to be related in a formal way to cost overhead. The purpose of this step is to express the whole model in the form of equations.

Process

To operationalize the model, we have to convert the relationships and the estimates of their magnitudes into a set of equations. As discussed above, our basic model is designed to be additive. The cost overhead value on each factor is a sum of the value on its constituent variables. The values on the k factors are summed up to provide the overall cost overhead (in percentage of nominal cost).

The equations for obtaining that value for direct, two way, and three way interactions have been derived and are summarized in Figure 6. The left hand side of the equations captures the cost overhead for a given variable, with or without interactions. The right hand side is expressed in terms of what we know: the information in the multiplier forms (Figure 5) and the project questionnaires.

For deriving the equations, we use the simplifying assumption that there are linear relationships between the variables of our questionnaire and project cost overhead (CO):

$$CO(V) = aV + CO_{nom}$$

where a is the slope of the linear relationship between cost overhead and the variable V. CO_{nom} is the intercept and minimal value of CO(V). CO_{ext} is defined as the maximum value that CO(V) can take. The indices *nom* and *ext* stand for the *nominal* and *extreme* values of cost overhead due to V.

We have: $a = \frac{CO_{ext} - CO_{nom}}{3}$ since all our scales are four point defined as 0 .. 3. In some situations, CO_{nom} corresponds to the case where the variable has no effect on cost overhead and may be set to 0.

Direct Relationship:

$$HCO_{f,q}(V_{f,q}) = V_{f,q} \times \frac{HCO_{f,q}(3)}{3}$$

Two way interaction:

$$HCO_{f,q}(V_{f,q}, V_{w,r}) = \frac{HCO_{f,q}(3, V_{w,r})}{3} \times V_{f,q}$$

where:

$$HCO_{f,q}(3, V_{w,r}) = \frac{(HCO_{f,q}(3, M_{w,r}) - HCO_{f,q}(3, 0))}{3}$$

Three-way interaction:

$$\mathrm{HCO}_{f,q}(3, V_{w,r}, V_{y,s}) = \beta_0 + (\beta_1 \times V_{w,r}) + (\beta_2 \times V_{y,s}) + (\beta_3 \times V_{w,r} \times V_{y,s})$$

where:

$$\beta_0 = \mathrm{HCO}_{f,q}(3, 0, 0)$$

$$\beta_1 = \frac{\mathrm{HCO}_{f,q}(3, V_{w,r}, 0) - \mathrm{HCO}_{f,q}(3, 0, 0)}{3}$$

$$\beta_2 = \frac{\mathrm{HCO}_{f,q}(3, 0, V_{y,s}) - \mathrm{HCO}_{f,q}(3, 0, 0)}{3}$$

$$\beta_3 = \left(\frac{\mathrm{HCO}_{f,q}(3, M_{w,r}, M_{y,s}) - \mathrm{HCO}_{f,q}(3, M_{w,r}, 0) - (\mathrm{HCO}_{f,q}(3, 0, M_{y,s}) + \mathrm{HCO}_{f,q}(3, 0, 0))}{9}\right)$$

Figure 6: Equations of the cost overhead estimation model.

The assumptions of the models presented in Figure 6 are:

1. All of the variables are orthogonal, hence justifying an additive model. The variables have been reviewed several times by senior project manager to ensure that they are orthogonal.
2. The scales used to measure each of the variables are interval or approximately interval. The scores assigned on the four point scales of the project questionnaire are assumed to be approximately equidistant. As explained earlier, this assumption is justified for our scales.
3. Each variable, as measured in the project data questionnaire, is assumed to have a linear relationship with cost overhead. For this simplifying assumption to be reasonable, assumption 2 above has to hold as well. The assumption of linear relationships is commonly made in empirical research when there is a lack of information or knowledge otherwise. This assumption has worked reasonably well in practice however.
4. Interaction effects between variables are also assumed to be linear. This is a common assumption in statistics, e.g., in regression analysis [12].

Outputs

At the end of this step, a quantitative cost overhead estimation model should have been produced. This model is expressed as the sum of triangular distributions, one for each variable. The parameters of these distributions are the HCO, MCO, and LCO equations for each variable.

Case Study

For our case study the complete model was implemented on a spreadsheet.

3.6 Estimating Cost Overhead

Objectives

For the model to be used, Monte Carlo simulation techniques are necessary to sample from the triangular distributions. This procedure is now explained.

Process

To estimate cost overhead, the first thing is to obtain responses on all the questions in the past project questionnaire. These values are translated into parameters of triangular distributions as explained in the previous step. We then run a Monte Carlo simulation. During the simulation we sample from each of the triangular distributions, and then sum the individual values to obtain a cost overhead estimate. This is repeated 1000 times. After 1000 iterations we have a distribution of cost overhead for the project. To get a point estimate of the cost overhead, one can take the mean of the distribution.

In principle, given that the model as described above is additive, the cost overhead distribution can be derived analytically without resort to simulation. Using a simulation approach has a number of advantages, however:

1. It provides the possibility to easily consider statistical associations amongst the factors in the causal model. This is important in order to generate, during the simula-

tion process, realistic joint distributions of factors. This will affect the Monte Carlo simulation process and the resulting cost overhead distribution [27].

2. It facilitates the aggregation of several experts' opinions (i.e., multipliers' distributions) in a flexible manner, e.g., by considering different weightings.
3. It makes the implementation of the model very easy to change should it be decided in the future to use a different distribution (e.g., a BetaPERT or some other based on stronger empirical evidence).
4. A project manager can assign probabilities to each of the 4 response categories on each question instead of selecting a unique response. This is particularly useful when using the model at the start of the project where only incomplete information is available. .

When doing the simulation we need to aggregate or combine the multipliers of many experts. Assuming we give equal weight to each of the experts, for each simulation run, one expert is selected at random with all experts having equal probability of being selected. Given that probabilities are equal, over 1000 simulation runs, each expert will be selected an approximately equal number of times as the other experts. Once an expert is selected, his/her multipliers are used to compute the overhead cost.

Outputs

The output of the simulation is a distribution of cost overhead. One can select, for example, the mean of this distribution as the estimated value of cost overhead for a project.

Case Study

The output of a simulation run is the cost overhead distribution. This can be produced for each project for which the project data questionnaire was filled up by the project manager. For example, one project manager retroactively filled up the questionnaire. He was asked to respond in terms of what was known at the beginning of the project. This was project of 140 KLOC developed in a 4GL that consumed 75 man months. The mean of the distribution was a cost overhead of 214% above that of a nominal project.

4 Validation of Model

4.1 Cost Overhead and Productivity

Size and effort data were collected retroactively for recently completed projects. These projects were considered to be representative of the types of projects that are conducted within the organization. All of the projects that we collected data on were considered to be "successful", that is they had been completed, were fully operational, and were deemed to be of acceptable quality. Although these criteria did not lead to the elimination of any project in our case study, such a selection is in general necessary in order to use a baseline of comparable projects, with consistent and meaningful size measurement. Size was measured in terms of non-comment lines of code excluding code produced by code generators. In addition, project managers filled up the questionnaire for their respective projects in order to obtain the data to feed our cost overhead model.

Subsequently, we obtained the cost overhead estimate for each project as explained above and determined the correlation between the cost overhead estimate and productivity. If the correlation is deemed significant and sufficient, then the validation is deemed successful. It is important to note that, despite the fact that this step requires project data, it only looks at a bivariate relationship and therefore does not require as much data as the construction of a data-driven, multivariate cost model. Recall this is a major objective of our method.

We only had complete data for six projects. Productivity was measured in terms of LOC per man-month. All six were development projects. We used the Spearman rank correlation coefficient [22] to evaluate the relationship between productivity and the mean cost overhead estimate. The mean cost overhead estimate was computed for each project from its cost overhead distribution. The correlation was found to be -0.886, and statistically significant at the 0.05 one-tailed alpha level. This indicates a high extent of validity of the cost overhead estimate.

4.2 Validation of Cost Estimates

To produce a cost estimate, we must estimate the beta parameters in eqn. 1. This can be done by fitting a line that minimizes the sum of squares criterion. In our case this criterion was adequate as there were no extreme observations in our data set, so a more robust fitting criterion was not necessary. To evaluate the accuracy of estimating cost using this approach, we performed a v-fold cross-validation. We fitted a line on five projects, and estimated the sixth. This was repeated six times for each of the projects. The average Absolute Relative Error was 0.09, which is a very good accuracy. This basically means that on average the model will over/underestimate by 9% of the actual. While the data set we used for this validation is small, the results are very encouraging indeed.

5 Benchmarking and Risk Assessment

In this context, benchmarking is the activity that consists of assessing how a given project compares to "typical" projects in the organization with respect to their cost overhead, i.e., whether it is a particularly difficult project or not. Such benchmarks can be used to help managers decide about the selected team composition and experience or even about the contracting of the project.

Risk assessment focuses on predicting the probability of being over budget or, similarly, a given budget overrun. Regardless of the way the budget has been set, this is a relevant activity which could lead to actions to either alleviate the risk (e.g., increase the planned budget, allocate more experienced people) or prepare contingency plans (e.g., emmergency budget for several projects).

Although benchmarking and risk assessment look very different in their purposes, they can be implemented using similar principles and may lead to similar decisions. In order to perform benchmarking or risk assessment, we have to set some CO thresholds.

For benchmarking, thresholds are defined to characterize the CO distribution (e.g., median, quartiles) from a set of representative, completed historical projects. For example, two thresholds can be defined as follows:

- The first threshold is that of a "typical" project. For example, in our case study, this is defined as the median

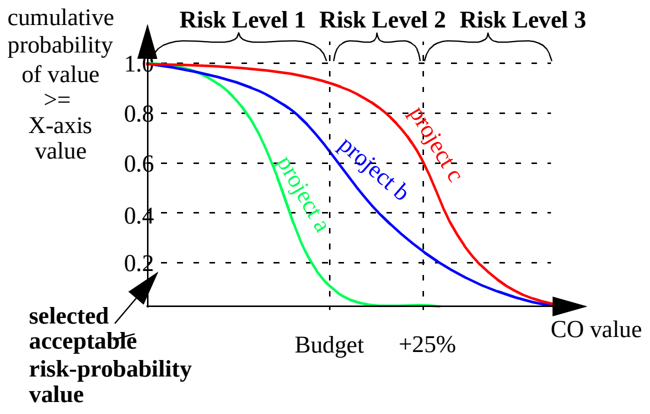

Figure 7: Risk assessment levels.

CO value for the nine projects on which we had project questionnaire data. This was some value T_t, and implies that 50% of the projects will have a mean CO value greater than T_t.

- Another threshold is that for the "majority" of projects. For the sample projects we define the majority as some value T_m, which is the upper quartile of the mean of CO values. The upper quartile has 75% of the projects' CO means below it.

Regardless of the exact thresholds, the comparison of a project against a representative baseline, e.g., the "typical" and "majority" thresholds above, tells in relative terms how difficult the project will be.

With respect to risk assessment, thresholds are determined by computing the CO values which would correspond, for example, to the planned budget or a 25% budget overrun. These CO values can be easily obtained by computing the productivity corresponding to the budget, or any overrun, and then use the productivity model to derive CO. We use the above thresholds to delimit *risk levels* (see Figure 7). Our two thresholds define three risk levels. Risk level 1 defines the lowest (below budget), risk level 2 moderate (below 25% budget overrun), and risk level 3 the highest risk (more than 25% budget overrun). A specific set of corrective and/or preventative actions should be associated with each risk level, except the lowest one (risk level 1). The higher the risk level, the more consequential (and likely costly) the actions.

To determine the probability of a particular project's CO value being greater than a threshold, we construct the cumulative probability distribution of the project's CO values from the 1000 Monte Carlo simulation runs by using eqn. 3. Such a distribution is then used to determine what actions are to be taken, if any. As an example, in this study, we use as a decision criterion a maximum acceptable probability $P_M = 0.2$ for the project's CO value to lie in a given benchmark or risk interval. For bencharking, a CO value above P_M for a given benchmark interval will determine that the project belongs to the corresponding level of difficulty (e.g., above typical, above majority). Regarding risk assessment, a CO value above P_M for a given risk interval will trigger the corresponding risk level's corrective or preventative actions.

For both benchmarking and risk assessment, threshold values would be determined by the most experienced project managers in conjunction with the quality assurance staff. They should be revised as more experience with the use of the models for benchmarking and risk assessment is gained. It should be remembered that acceptable risk is a business decision, and should reflect the objectives of the project to be analyzed and the business strategy of the organization.

In the hypothetical example in Figure 7, it can be seen that the "low risk" example project has a probability of less than 0.2 of having a cost overhead equal to or exceeding that of the planned budget and 25% overrun thresholds. Therefore, it is considered as being risk level 1. On the other hand, the "high risk" project has a probability greater than 0.2 of exceeding the budget or even a 25% budget overrun. Therefore it is considered to be at risk level 3.

As a benchmark example from our study, let us consider the project that had a mean cost overhead of 214%. This project had a probability of getting an CO value greater than the "typical" project of almost 0.6. When considering the "majority" of projects, the probability of exceeding them was approximately 0.3. Therefore, this may be considered a project of "above majority" difficulty level.

Regarding risk assessment, and in order to focus these actions, it would also be useful to know which variables are the lead causes of high risk. This can be achieved by ranking the variables by their mean $MCO_{f,q}$ values across the seven project managers who provided multipliers using the equations similar to those in Figure 6. This ranking indicated that the variable "The requirements were not well understood by all parties (developers and customers) at the beginning of the project" was by itself adding 22% to the cost overhead estimate, and consequently, was adding approximately 5 man-months to the project taken as example here.

Based on this information, the project manager can look at how successful risk level 3 projects in the past have dealt with weak understanding of requirements, and take similar actions.

6 Conclusions

We presented a method for COst estimation, Benchmarking, and Risk Assessment: COBRA. This method has shown to be convenient and low cost when an organization needs to develop local cost and risk models and is not able to collect or retrieve a large set of project data. For example, the case study presented above took approximately 2 man-months of interviewing and analysis effort. We have also shown how project manager expertise can be collected, refined, modeled, validated, and used for cost overhead estimation and cost risk benchmarking and assessment. We have illustrated how the uncertainty associated with expert opinion can be modeled, integrated in the cost overhead model, and used through Monte Carlo simulation. Our resulting cost estimation and risk assessment models are operational and their construction is repeatable through a well-defined process. Our case study has shown good initial results on actual projects, thus demonstrating the feasibility of such an approach.

Future work includes the use of the method in other application domains than MIS and its adaptation to the cost risk assessmentRisk assessment of maintenance releases.

7 Acknowledgements

This work was supported in part by software design & management (sd&m). We are thankful to all the sd&m employees involved in the case study who have made this work possible.

References

[1] J. Bailey and V. Basili: “A Meta-Model for Software Development Resource Expenditures”. In *Proceedings of the International Conference on Software Engineering*, pages 107-116, 1981.

[2] R. Banker and C. Kemerer: “Scale Economies in New Software Development”. In *IEEE Transactions on Software Engineering*, 15(10):1199-1205, 1989.

[3] B. Boehm: *Software Engineering Economics*. Prentice-Hall, 1981.

[4] B. Boehm, E. Horowitz, R. Selby, and C. Westland: *COCOMO 2.0 User’s Manual: Version 1.1*, University of Southern California, 1994.

[5] Lionel C. Briand, Victor R. Basili and William M. Thomas: “A Pattern Recognition Approach for Software Engineering Data Analysis”, In *IEEE Transactions on Software Engineering*, 18(11):931-942, 1992.

[6] M. Evans, N. Hastings, and P. Peacock: *Statistical Distributions*. John Wiley & Sons. Inc., 1993.

[7] R, Fairley: “Risk Management for Software Projects”. In *IEEE Software*, pages 57-67, 1994.

[8] S. Grey: *Practical Risk Assessment for Project Management.* John Wiley & Sons Ltd., 1995.

[9] F. Heemstra: “Software Cost Estimation”. In *Information and Software Technology*, 34(10):627-639, October 1992.

[10]J. Hihn and H. Habib-Agahi: “Cost Estimation of Software Intensive Projects: A Survey of Current Practices”. In *Proceedings of the International Conference on Software Engineering*, pages 276-287, 1991.

[11]M. Host and C. Wohlin: “A Subjective Effort Estimation Experiment”. In *Proceedings of the Conference on Empirical Assessment in Software Engineering* (EASE-97), 1997.

[12]J. Jaccard, R. Turrisi, and C. Wan: *Interaction Effects in Multiple Regression*. Sage Publications, 1990.

[13]C. Kemerer: “An Empirical Validation of Software Cost Estimation Models”. In *Communications of the ACM*, 30(5):416-429, May 1987.

[14]K. Kinsala: “Integrating Risk Assessment with Cost Estimation”. In *IEEE Software*, pages 61-67, May/June 1997.

[15]B. Kitchenham: “Empirical Studies of Assumptions that Underlie Software Cost Estimation Models”. In *Information and Software Technology*, 34(4):211-218, 1992.

[16]R. Kusters, M. van Genuchten, and F. Heemstra: “Are Software Cost-Estimation Models Accurate?”. In *Information and Software Technology*, 32(3):187-190, 1990.

[17]A. Lederer and J. Prasad: “Nine Management Guidelines for Better Cost Estimating”. In *Communications of the ACM*, 35(2):51-59, 1992.

[18]S. Mohanty: “Software Cost Estimation: Present and Future”. In *Software-Practice and Experience*, 11:103-121, 1981.

[19]H. Rubin: “A Comparison of Cost Estimation Tools”. In *Proceedings of the International Conference on Software Engineering*, 1985.

[20]H. Rubin: “Software Process Maturity: Measuring its Impact on Productivity and Quality”. In *Proceedings of the International Conference on Software Engineering*, pages 468-476, 1993.

[21]R. Scott and D. Simmons: “Programmer Productivity and the Delphi Technique”. In *Datamation*, pages 71-73, May 1974.

[22]S. Siegel and N. Castellan: *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, Inc., 1988.

[23]K. Srinivasan and D. Fisher. “Machine LearningApproaches to Estimating Software Development Effort”, In *IEEE Transactions on Software Engineering*, 21(2):931-942, 1995.

[24]P. Spector: *Summated Rating Scale Construction.* Sage Publications, 1992.

[25]P. Spector: “Ratings of Equal and Unequal Response Choice Intervals”. In *Journal of Social Psychology*, 112:115-119, 1980.

[26]P. Spector: “Choosing Response Categories for Summated Rating Scales”. In *Journal of Applied Psychology*, 61(3):374-375, 1976.

[27]D. Vose: *Quantitative Risk Analysis: A Guide to Monte Carlo Simulation Modelling*. John Wiley & Sons, Inc., 1996.

[28]C. Wrigley and A. Dexter: “Software Development Estimation Models: A Review and Critique”. In *Proceedings of the ASAC Conference*, University of Toronto, 1987.

Appendix A: Past Project Questionnaire

A Past Project Data Questionnaire – Final Version

A.1 About This Questionnaire – How to Answer Questions

The objective of this questionnaire is to collect information about a single project that you have been involved in at sd&m. A single project is a system development that has a single contract with a customer. Therefore, for example, development efforts of the same system that had three consecutive contracts would be considered as three different projects.

Select a project that you are very familiar with, for example, a project that you have managed or a project on which you were the lead analyst. Answer the questions with reference to that project. Also, please do not switch projects while you are filling out the questionnaire.

Please answer all of the questions in the questionnaire unless there is an arrow indicating that you should skip one or a series of questions.

The information we collect about projects using this questionnaire is very critical for us to validate and fine tune the cost estimation decision model that we are developing, so it is very important that you answer all of the questions. We would prefer if you rather give your best guess than leave questions blank.

A.1.1 When in the Project

Some of the questions concern situations that existed at the beginning of the project, and some concern events during the project. For every question, it will be made clear to which point in the project we want your answer to refer to.

For each question, please check at which point in the project it is referring to and answer accordingly.

A.1.2 Question Types

There are two general types of questions in this questionnaire: questions that have a Likert-type answering scale and questions with factual answers:

Likert-type Scale

A *Likert-type scale* consists of a statement, and a number of response categories that the respondent chooses from. The types of response categories that we use here are *frequency*, *evaluation*, and *agreement* (these are very commonly used types of scales /13/). Frequency type scales ask the respondents about how many times the activity described in the statement would happen. Evaluation type scales are used in this document to rate the capability of a key person on the project along a good-bad dimension. Agreement type scales ask respondents about their extent of agreement to the statement. This statement usually is a characterization of a project.

The way we eventually use these scales assumes that the intervals between each of the categories are equal. Using the work of Spector /13/, we have used phrases that are approximately *conceptually* equally spaced for both the frequency and evaluation scales.

The agreement scale is bipolar and symmetrical, which should make responding easier. We have employed a very commonly used agreement scale from which we can arrive at approximately equally spaced intervals.

Factual Questions

Factual questions ask about facts concerning the project, such as the project name. It is critical that the answers to these questions are as accurate as possible. If necessary, please check previous files or other sources of information to ensure that the responses are as accurate as possible.

The factual questions are left until the end of the questionnaire so that, if necessary, you can collect extra information not readily available without disrupting the flow of answering the questionnaire.

A.1.3 Organization of Questions

Questions are organized so that questions that deal with a related set of issues are together under the same heading.

A.2 The Questionnaire

A.2.1 General Project Information

A.2.1.1 What is the name of the project or contract?

____

A.2.1.2 What was your position in this project?

- ❒ Project Manager
- ❒ Lead Analyst
- ❒ Programmer
- ❒ Other *(please specify below)*

____

A.2.2 Understanding and Consistency of Business Objectives for the Project and Product

Definition: The extent to which the business objectives for the project and product are clearly understood, and the understanding of objectives between the project team and the customer are consistent (i.e., no conflicts in their interpretation).

A.2.2.1 The business objectives for the project and the product were well defined at the start of the project:[1]

❒ Strongly Agree	<0>
❒ Agree	<1>
❒ Disagree	<2>
❒ Strongly Disagree	<3>

A.2.2.2 The business objectives for the project and product were documented at the start of the project:

❒ Strongly Agree	<3>
❒ Agree	<2>
❒ Disagree	<1>
❒ Strongly Disagree	<0>

A.2.2.3 The business objectives for the project and the product were understood by the development team at the start of the project:

❒ Strongly Agree	<3>
❒ Agree	<2>
❒ Disagree	<1>
❒ Strongly Disagree	<0>

A.2.2.4 The business objectives of the project and the product were understood by the customer at the start of the project:

❒ Strongly Agree	<3>
❒ Agree	<2>
❒ Disagree	<1>
❒ Strongly Disagree	<0>

A.2.2.5 There was one or more persons at the customer site who was/were clearly responsible and available for customer decision making:

❒ Strongly Agree	<3>
❒ Agree	<2>
❒ Disagree	<1>
❒ Strongly Disagree	<0>

A.2.2.6 Were there several customer departments involved in the project?

❒ NO	<1>
❒ YES	<2>

1. For each option of the multiple choice type answers, we have added a number in angular brackets which is an encoding of this option's value. The codes are used in the tables presented in Appendix I.

A.2.2.7 The customer departments had conflicting interests which had to be resolved:

❐ Strongly Agree	<3>
❐ Agree	<2>
❐ Disagree	<1>
❐ Strongly Disagree	<0>

A.2.3 Key Project Team Capabilities

Definition: The knowledge of key people on the project team (e.g., lead analyst, and project manager) about the application domain for the project, the process and documentation standards and common practices to be used on the project, the development platform and environment, and dealing with people.

A.2.3.1 At the start of the project, the familiarity with and comprehension of the application domain of the key people on the project:

❐ Inferior	<3>
❐ Unsatisfactory	<2>
❐ Satisfactory	<1>
❐ Excellent	<0>

A.2.3.2 At the start of the project, the familiarity with and comprehension of the platform to be used (e.g., programming language(s), Operating System, database management systems) of the key people on the project:

❐ Inferior	<3>
❐ Unsatisfactory	<2>
❐ Satisfactory	<1>
❐ Excellent	<0>

A.2.3.3 At the start of the project, the familiarity with the type of system architecture used (e.g., client-server, Internet JAVA applications) of the key people on the project:

❐ Inferior	<3>
❐ Unsatisfactory	<2>
❐ Satisfactory	<1>
❐ Excellent	<0>

A.2.3.4 At the start of the project, the familiarity with and comprehension of the software development environment (e.g., compiler, code generator, CASE tools) of the key people on the project:

❐ Inferior	<3>
❐ Unsatisfactory	<2>
❐ Satisfactory	<1>
❐ Excellent	<0>

A.2.3.5 At the start of the project, the ability to communicate easily and clearly with the others (e.g., good interviewing skills and other information gathering techniques, good verbal communication skills, ability to lead people) of the key people on the project:

❒ Inferior	<3>
❒ Unsatisfactory	<2>
❒ Satisfactory	<1>
❒ Excellent	<0>

A.2.3.6 At the start of the project, the knowledge and experience of the software development process and techniques to be used during the project (e.g., functional and/or object modeling techniques, testing techniques, and conducting a cost/benefits analysis) of the key people on the project:

❒ Inferior	<3>
❒ Unsatisfactory	<2>
❒ Satisfactory	<1>
❒ Excellent	<0>

A.2.3.7 At the start of the project, the knowledge and experience of the documentation standards to be used during the project (e.g., modeling notation, and requirements document structure and content) of the key people on the project:

❒ Inferior	<3>
❒ Unsatisfactory	<2>
❒ Satisfactory	<1>
❒ Excellent	<0>

A.2.4 Customer Participation

Definition: The extent to which the customers are efficiently and promptly performing some of the development activities themselves, providing information, and/or reviewing project documents.

A.2.4.1 Customers provided information to the project team (e.g., during interviews, when given questionnaires by the project staff, when presented with a “system walk-through”, and/or when they are asked to provide feedback on a prototype):

❒ Rarely	<3>
❒ Infrequently	<2>
❒ Occasionally	<1>
❒ Most of the Time	<0>

A.2.4.2 The adequacy of information provided by the customer during the project (e.g., during interviews, when given questionnaires by the project staff, when presented with a “system walk-through”, and/or when they are asked to provide feedback on a prototype):

❒ Inferior	<3>
❒ Unsatisfactory	<2>
❒ Satisfactory	<1>
❒ Excellent	<0>

Note: Information provided is considered excellent if it was accurate and complete, and was provided in an efficient and timely manner.

A.2.4.3 Customers reviewed the work done by the project team:

❒ Rarely	<3>
❒ Infrequently	<2>
❒ Occasionally	<1>
❒ Most of the Time	<0>

A.2.4.4 The adequacy of the reviews done by the customer:

❒ Inferior	<3>
❒ Unsatisfactory	<2>
❒ Satisfactory	<1>
❒ Excellent	<0>

Note: Reviews were excellent if they were accurate and complete, and done in an efficient and timely manner.

A.2.5 Mixed Teams

A.2.5.1 Were customers actively involved in the project (i.e., members of a mixed project team)?

❒ NO	<1>
❒ YES	<2>

A.2.5.2 The sd&m staff and the customers on the project team had worked together in the past on previous project(s) / contract(s):

❒ Rarely	<3>
❒ Infrequently	<2>
❒ Occasionally	<1>
❒ Most of the Time	<0>

A.2.5.3 Customers participated in the development activities (e.g., writing the requirements specifications, developing screens and screen layouts, developing test data specifications, liaising between the development team and the users, and/or participate in creating the user manual):

❒ Rarely	<3>
❒ Infrequently	<2>
❒ Occasionally	<1>
❒ Most of the Time	<0>

A.2.5.4 The quality of the customer's output (e.g., documents produced, code and screens developed) during the project:

❒ Inferior	<3>
❒ Unsatisfactory	<2>
❒ Satisfactory	<1>
❒ Excellent	<0>

Note: Outputs are of excellent quality if they were accurate and complete, and were produced in an efficient and timely manner.

A.2.6 Requirements Volatility

Definition: The extent to which the agreed upon requirements are expected to change over time during the project.

A.2.6.1 Requirements were understood by all parties at the beginning of the project:

❐ Strongly Agree	<0>
❐ Agree	<1>
❐ Disagree	<2>
❐ Strongly Disagree	<3>

A.2.6.2 Interface specifications to software that is being developed in parallel changed during the project:

❐ Strongly Agree	<3>
❐ Agree	<2>
❐ Disagree	<1>
❐ Strongly Disagree	<0>

Note: If no parallel development takes place please check “Strongly Disagree”.

A.2.6.3 The customer organization was not able to make decisions or was not fully committed to the decisions it made during the project:

❐ Strongly Agree	<3>
❐ Agree	<2>
❐ Disagree	<1>
❐ Strongly Disagree	<0>

A.2.6.4 The turnover of the customer management (or those on the customer side who define the requirements) was high during the project:

❐ Strongly Agree	<3>
❐ Agree	<2>
❐ Disagree	<1>
❐ Strongly Disagree	<0>

A.2.6.5 The tasks that shall be supported by the to be developed software at the customer site were undertaking numerous changes during the project:

❐ Strongly Agree	<3>
❐ Agree	<2>
❐ Disagree	<1>
❐ Strongly Disagree	<0>

A.2.6.6 The system functionality was completely new for the customer:

❐ Strongly Agree	<3>
❐ Agree	<2>
❐ Disagree	<1>
❐ Strongly Disagree	<0>

A.2.7 Development Schedule Constraints

Definition: The extent to which a reasonable project schedule is compressed without changing any of the stated requirements.

A.2.7.1 What was the cutback from the reasonable schedule (schedule compression) if a reasonable schedule is considered to be 100%?

❐ 1% - 5%	<1>
❐ 6% - 10%	<2>
❐ 11% - 20%	<3>
❐ 21% - 50%	<4>

A.2.8 Meeting Reliability Requirements

Definition: The amount of extra attention beyond what is stipulated in the sd&m common practices that is necessary to meet the reliability requirements for the part of the system developed by sd&m.

A.2.8.1 Any risk associated with an operational failure that would have unacceptably adverse economic, safety, security, and/or environmental consequences can be reduced or eliminated without extra attention beyond following the common sd&m development practices:

❐ Strongly Agree	<0>
❐ Agree	<1>
❐ Disagree	<2>
❐ Strongly Disagree	<3>

A.2.9 Meeting Usability Requirements

Definition: The amount of extra attention beyond what is stipulated in the sd&m common practices that is necessary to meet the usability requirements for the part of the system developed by sd&m.

A.2.9.1 A detailed task analysis was necessary in order to meet the usability requirements for this project:

❐ Strongly Agree	<0>
❐ Agree	<1>
❐ Disagree	<2>
❐ Strongly Disagree	<3>

A.2.9.2 The usability requirements do exceed what can be achieved by just following the common sd&m practices, user interface development guidelines, and available tools and libraries:

❐ Strongly Agree	<3>
❐ Agree	<2>
❐ Disagree	<1>
❐ Strongly Disagree	<0>

A.2.10 Meeting Performance Requirements

Definition: The amount of extra attention beyond what is stipulated in the sd&m common practices that is necessary to meet the performance (i.e., response time, execution time, and memory usage) requirements for the part of the system developed by sd&m (and that is not explicitly captured by an extra work package).

A.2.10.1For a significant part of the system functions, response time remains acceptable under all conditions and performance degradation under high load is insignificant when they are developed following the common sd&m practices:

❒ Strongly Agree	<0>
❒ Agree	<1>
❒ Disagree	<2>
❒ Strongly Disagree	<3>

A.2.11 Disciplined Requirements Management

Definition: The process that is needed for managing changes in requirements beyond what is considered the sd&m common process.

A.2.11.1 Requirements were well documented at the beginning of the project:

❒ Rarely	<3>
❒ Infrequently	<2>
❒ Occasionally	<1>
❒ Most of the Time	<0>

A.2.11.2 An explicit impact analysis was performed for requirements changes:

❒ Rarely	<3>
❒ Infrequently	<2>
❒ Occasionally	<1>
❒ Most of the Time	<0>

A.2.11.3 Requirements changes were prioritized based on their importance for the customer and project cost and schedule impacts:

❒ Rarely	<3>
❒ Infrequently	<2>
❒ Occasionally	<1>
❒ Most of the Time	<0>

A.2.11.4 Requirements were traceable to other work products (e.g., design documents, code, test cases, and project plans):

❒ Rarely	<3>
❒ Infrequently	<2>
❒ Occasionally	<1>
❒ Most of the Time	<0>

Note: This is usually done by cross-referencing or mapping tables.

A.2.11.5 Commitments with the customer were explicitly renegotiated and documented based on requirements changes:

❒ Strongly Agree	<3>
❒ Agree	<2>
❒ Disagree	<1>
❒ Strongly Disagree	<0>

A.2.12 Software Size

Definition: We measure the size of the software by non-commented lines of code. We allow for the use of up to three different programming languages on a project. Please provide the data for each programming language that may have been used on the project.

Definition: Where code generators have been used, please enter the source line of code size *not* the generated line of code size.

Programming Language 1:

A.2.12.1 What was the programming language used? *(Please specify below)*

____

A.2.12.2 Non-Commented Lines of Code:

New Code: ____________________________ Lines of Code

Modified Code: ________________________ Lines of Code

Old (Reused) Code: _____________________ Lines of Code

Programming Language 2:

A.2.12.3 What was the programming language used? *(Please specify below)*

____

A.2.12.4 Non-Commented Lines of Code:

New Code: ____________________________ Lines of Code

Modified Code: ________________________ Lines of Code

Old (Reused) Code: _____________________ Lines of Code

Programming Language 3:

A.2.12.5 What was the programming language used? *(Please specify below)*

____

A.2.12.6 Non-Commented Lines of Code:

New Code: ___________________________ Lines of Code

Modified Code: _______________________ Lines of Code

Old (Reused) Code: ____________________ Lines of Code

A.2.13 Project Effort Information

Please specify the total project effort for the Realization Phase of the project only (i.e., excluding the Specification Phase). The Realization Phase includes all activities from design to installation. Please **include** all of the following activities in the total effort:

- Design
- Coding
- Testing (e.g., integration testing and acceptance testing)
- Installation
- Rework during the Realization Phase
- Project Management
- Quality Assurance
- Configuration Management
- Software Engineering Environment Support
- Meetings
- Training and Integration of New Staff

Please **exclude** the following activities in the total effort:

- Travel
- Rework and Changes After Installation

A.2.13.1 Please specify the total effort for the realization phase:

Effort: ___________________________ man-months