Development of an Effort Estimation Model – A Case Study on Delivery Projects at a Leading IT Provider within the Electric Utility Industry

Teodor Sommestad, Joakim Lilliesköld

Department of Industrial Information and Control Systems
Royal Institute of Technology (KTH),
Osquldas väg 12, S-100 44
Stockholm, Sweden

Abstract. When projects are sold with fixed prices, it is utterly important to quickly and accurately estimate the effort required to enable an optimal bidding. This paper describes a case study performed at a leading IT provider within the electric utility industry, with the purpose of improving the ability to early produce effort estimates of projects where standard functionality is delivered. The absence of reliable historic data made expert judgment the only appropriate foundation for estimates, with difficulties of quickly develop estimates and reuse or modify estimates already made. To overcome these troubling issues, the expert estimates were incorporated into a model where they and the factors influencing them are traceable and readily expressed. The model is based on decomposition of projects and bottom-up estimation of them, where impact of relevant variables is estimated by assessing discrete scenarios. It provides quick and straightforward means of developing estimates of the decomposed elements and whole projects in various circumstances, where not only expected effort is considered, but the uncertainty of the individual estimates is visualized as well. This does together with the traceability enable the estimates produced by the model to be assessed, analyzed and refined as more details of the project is known.

I. Introduction

A substantial share of the software delivery projects carried out within the industry deals with integration and adaptation of pre-developed software packages. These implementation projects differ from the software development project in the sense that none or little effort has to be spent on code development, and naturally the nature of work performed in the projects differs. Even though the repetitiveness of implementation projects using pre-developed software may increase the possibility to predict the effort required in a project, the demands on running and managing projects effectively is increased as well. To run and sell projects with a profit margin you need the capability to predict costs and risks associated with potential undertakings.

Accurate estimates on the effort required to carry through a project enables the supplier to choose the most appropriate projects to compete for, and in the bidding on these choose a suitable level for the bid. Making early estimates on the effort required in projects is however extremely difficult [14]. The initial estimates need to be developed quickly, often based on uncertain assumptions. There is rarely time, resources or information available to thoroughly perform the suggested steps [20] of planning and costing, such as a thorough work breakdown, activity sequencing and network analysis. The estimation approaches feasible in early phases are usually parametric or analogous estimation [8][13][18] and to support the estimators, numerous estimation models for software projects has been proposed. The parametric models are based on relationships between parameters and effort whereas analogous models are based on a database containing previously performed activities and derives the estimate based on the effort of these.

Although the research conducted on estimation models for software projects is fairly extensive, the use of formal models in the software industry seems to be very limited compared to the use of expert judgment. In fact Jørgensen [11] reports that none of the surveys found on effort estimation approaches has shown the use of formal models as the dominant strategy in development of effort estimates in software projects. As Jørgensen [11] states there could be many reasons for why the industry relies on expert judgment, such as estimators feeling uncomfortable using models they do not fully understand or the inability of models to outperform expert judgment.

For an estimator involved with the bidding on implementation projects of pre-developed software there are at least two concerns regarding these models. Firstly, the models are data intensive and thereby inaccurate in early project phases [14]. Secondly, the models are developed for supporting estimation of software development projects, not the implementation and customization of pre-developed software [8][26]. The modest need of code development in projects based on pre-developed software is unaligned with the use of size measures such as lines of code, which is used in estimation models focused on software development projects [8][26].

In a study of ERP projects Francalanci [8] examined the effort drivers of implementation projects of ERP systems, which normally is based on pre-developed software. Francalanci found that the number of modules implemented, the number of users involved, and the size of the organizations is correlated to implementation effort of ERP-projects. Furthermore, Francalanci [8] found that the number of sub-modules of a module correlates with the effort of implementing a module. In another study Stensrud [26] evaluated the applicability of software estimation approaches in effort estimation of ERP projects, The conclusion from this evaluation was that the only software development approach that makes real sense in the context of pre-developed software was regression analysis, which like most models requires large amounts of data on past achievements for the analysis to yield reasonable estimates.

As noted earlier, there are several approaches and methods available to predict the effort and cost of software projects. Which effort estimation approach that generally provides the highest accuracy course debatable, and it could depend on the circumstances [23]. Research made on effort estimation models for projects implementing pre-developed software is compared to research on effort estimation models for software development projects, small to the number [26]. Projects involving pre-developed software differ from software development projects and consequently the models for predicting software development effort are in many senses inapplicable to pre-developed software projects.

This paper presents a case study performed on a leading developer and supplier of pre-developed software within the electric utility industry. The investigated organization, like many other organizations, lacked reliable historic data to base predictions on. Instead experience, judgement and negotiations served as basis for developing estimates. The study investigated the needs of effort estimation support in the bidding and planning process and based on the requirements surfaced from this investigation a model for effort estimation is suggested.

To bridge the gap between sales, management and project staff in the estimation process, a simplistic model was developed that could be based on estimates made by experts and specialists of different project domains. The model thus gathered expert knowledge from different parts of the organization and formalized it into a model. This provided individual with without deep insights about the effort of activities performed during a project to quickly develop initial estimates. Based on these initial estimates, refinements could be made based on either project specific circumstances or additional information about the project. The contribution from this case study is firstly a number of requirements placed on an effort estimation model that is to be used in the context of pre-developed software implementation. Secondly, an effort estimation model is proposed to fulfil these requirements, and thereby suit the context of projects implementing pre-developed software.

The remainder of this paper is organized as follows. First the case study is described together with the investigated organization, the projects and the effort estimation procedure deployed before and during projects. Following this, in section three, the requirements made on an estimation model from the investigated organization is presented. In section four a model is proposed to satisfy these requirements and some figurative examples of the implemented models output is presented. In the fifth and last section, a summary and discussion is provided.

II. The Case Study

This paper is based on a case study performed during 2007 with the purpose to investigate the need for support in development of effort estimates in early phases of projects implementing pre developed software. Within this study, the requirements placed on a model to provide support in the estimation process was identified and a model was proposed and implemented to meet these requirements. The organization investigated in the case study is a fortune 500 engineering enterprise, supplying the utility industry with SCADA/EMS solutions for power generation, transmission and distribution. The software product comprises of a number of modules with corresponding sub-modules and in a project a subset of these is implemented and integrated.

The data collection method used to answer the case study question is primarily interviews, however, documentation and demonstrations of the system have been used as well. A dozen interviews were conducted with engineers, managers and sales personnel in a semi-structured manner. Managers and sales personnel where interviewed on subjects relating to the project in general, but focused on bidding and planning of projects and how estimates where developed and applied during these processes. Engineers selected by managers on the criteria of high experience of the projects where interviewed on similar issues. These interviews did however focus more on the impact of different variables on effort required in projects. When the model was implemented, the estimates was developed using the wideband Delphi process [2]. Nine individuals, with various background and expertise participated in the estimation process and formed groups according to their system domain knowledge. In these groups the estimates where developed to serve as a basis for the model.

A. The Implementation Projects

The scope of studied projects varies from the delivery of hardware and software to the design and construction of large scale turn-key installations. Systems could in some cases partially comprise of software products supplied and licensed by other software developers, but is in essentially built on a in-house developed software solution. To enable a high degree of adaptability of the product and enabling customer to select the desired functionality, the product is modularized into modules on a higher level, and more or less independent function packages on a lower level. Customer specific requirements in a project might require customizations that involve modifications of the software or development of new functionality. However, the bulk of the effort in the project concerns the implementation, parameterization and integration of pre-developed software modules and functions. When deviations from standard functionality require development as a part of a project, this is often managed independently.

Projects comprise of two major parts, system integration in the factory, and work on the site where the system is to be installed. These two parts can be further divided into specific activities which before the delivery have the objective to implement a system solution that work in a test environment; and after the delivery and installation test and run the project against the power process. There are some variations on what work is performed during a system delivery, and the order in which it is performed. Many of the activities are executed partly parallel and it is difficult to make clear distinctions between when one activity ends, and the next starts. Depending on weather the system is tested using customer specific data or not, and if this data is applied in factory or just on site, some activities are executed in different locations. This, together with the fact that projects are unique and have a slightly different scope and prerequisites makes it difficult to provide a general description of each activity.

In the typical case however, the projects start with the creation of a work statement together with the customer where matters such as deadlines and responsibilities are stated in more detail. This is followed by the creation of a model of the power network and the integration of functions and modules in the test environment at the factory. Different types of system tests are performed sequentially and the part in factory is ended by a traditional factory acceptance tests. When the factory acceptance test has been approved and necessary corrections have been made commissioning is initiated and the system is moved to site and tuned in against the process. When installed, tuned in, and potential errors have been corrected a site acceptance test is carried out to finalize the delivery.

Since the software is principally developed when a project starts the work performed during a project is mainly about creating a correct model of the network, setting parameters, and integrating the selected function modules with each other. The integration itself rarely causes any significant difficulties; the possibility of an incorrect model of the power network could however create some problems. In the typical project the work with developing, correcting and testing system against the power network model therefore constitute a substantial part of the effort.

Although the projects in comparison with software development projects are fairly repetitive, the scope can vary greatly between projects. From the suppliers point of view there are in particular three variables setting the scope of projects: the function scope, the activities performed and the delivery model used for the implementation. Given a scope there is couple of factors that influence the effort required to complete it, for example factors like the size of the power network that is to be modeled.

The functions included in a delivery are an obvious driver of effort and although there are dependencies between functions, there are numerous possible function combinations. The activities performed in a delivery may vary; the customer could for example be responsible for certain activities, such as entering network data into the system. Another determinant of scope is the delivery model that is used for the project. The impact of different delivery models is more complicated than just excluding and including activities. The delivery model determines which responsibilities the supplier has and which responsibilities that lies on the customer. There are of course many possible ways of dividing responsibility between the parties, but essentially there are two delivery models that are used. In the first and most frequently used model the responsibility for a correct network model is on the supplier, which consequently is responsible for making the system work with this model. In the second delivery model the responsibility of modeling the network is placed on the customer and the supplier of the system only assures the functionality of the system is correct. Hence, issues regarding modeling and errors in functionality caused by flaws in the network model are not a part of the supplier's undertaking in this delivery model.

Besides the scoping variables mentioned above there are a number of other factors influencing the effort of a project. A handful of variables are of particular importance, for example issues relating to the data about the power network. Depending on the scope of the project, these variables have different impacts. Functions and activities do for example vary in data usage, and impact of variables relating to this will depend on the function included and the activities performed. Furthermore, the impact from these circumstances is dependent on whether the responsibility of modeling the power network is placed on the supplier or not. If the responsibility is placed on the customer, the significance of data quality is naturally of less from the supplier perspective. These variables could naturally relate to human and organizational factors as well, such as the experience, maturity and domain knowledge of the customer purchasing the system.

B. The Effort Estimation Procedure

In the bidding procedure preceding the execution of a project the issues of project scope are of course settled on a high level. Together with additional information on the project this information serve as a basis for the prediction of effort required. The bidding procedure naturally varies between projects, but do in general involve a number of iterative steps where the procurers provide requirement specifications and complementing information when clarifications are required by bidders. The number of suppliers is sometimes gradually narrowed down and the ones selected for latter stages are provided more detailed information about the project to enable more thorough negotiations to be carried out. The estimates of effort is naturally refined as more information is provided on the project and while initial estimates may need to be developed with very limited information, the estimates developed later in the bidding process and during the project could be based on a fairly exhaustive information about the project.

A bid manager is usually appointed to investigate the requirements specification. Based on this, and support from others with insight, he or she develops an estimate of the required effort. Since the organization holds a great deal of experience about working with standard functionality in the projects, the estimations made on final effort of this part of the project is fairly accurate as a whole. The development of an estimate is however quite cumbersome since bid manager do not necessarily have required experience of the circumstances and functionalities in the project up for bidding.

Expertise on different parts of the system is distributed among the staff and the bid manager needs to gather this experience to make as accurate predictions as possible. To capture this distributed knowledge and provide a basis for discussions in the estimation process, spreadsheet templates incorporating a very simple estimation model where used. The template model provided experience based estimates of required effort on functions and phases. It furthermore enabled users to manually alter the effort estimates provided by it if this was seen as appropriate considering the circumstances. Although it provided experience based estimates, the overall opinion was that the template did not produce estimates that were correct. They were neither correct on the projects as a whole, or on different tasks separately. One of the problems with the existing template was that it did not consider all the major variations that exist between projects when it comes to other factors than scope. It thereby provided estimates based on assumptions that were unclear and presumably wrong from case to case. Another problem was that it was unclear how the tool produced its estimates and if the estimates are up to date with the current productivity rate of the organization, which increases the ambiguity of the estimate even further.

The idea behind the existing template was to provide bid managers and others with experience based estimates of the different activities performed on different parts of the system during a project. This was perceived as good, not only since it collected relevant knowledge and enabled quick initial estimates to be derived, but also since it potentially could provide a solid base for meaningful discussions about the required effort. The inaccuracy and ambiguity of the estimates which the template provided did however hinder the template from providing the desired features, and thereby it did not provide any genuine support.

To provide the desired support the model would have to be able to not only provide estimates based on the scope of the project, but also based on factors affecting the effort within different scopes. These factors could, as mentioned above, for instance be the expected size of the power network or the maturity of the customer. The respondents described a handful of factors which were both relevant and possible to asses in the early phases of a project. Impact of such factors could sometimes be a multiple of five on the effort and if a model could capture the impact of these, the estimates produced by it would able to provide significantly higher accuracy.

Since the organization possess a solid experience and great knowledge about the relation between variations and efforts in projects the experience based estimates taking advantage of this information yielded estimates with sufficient accuracy. The use of data on past achievements and corresponding efforts during the estimation was however sparse. A time reporting system had been used the last couple of years to report the time spent on different functions and activities in the projects. The efforts reported did however only cover a dozen of projects and the efforts reported in these would not serve well as predictions to future projects. There are mainly two reasons for this: efforts on past projects are not likely to correspond to coming ones, and the reported effort is not accurate and detailed enough.

There are, according to respondents, reasons to believe that increased efficiency within the projects had made reports on effort of past projects outdated. Hence, the reported effort is likely to be higher than the effort needed in corresponding projects today. Something that obviously lowered the usability of reported efforts. The second reason for not basing the estimation model on archived reports of effort concerns its accuracy and reliability. Since the work in a project deals with the functions and activities in parallel, the borders between activities is indistinct, and the effort reported on specific work packages is therefore not very reliable. Furthermore, the project members do sometimes work in several projects simultaneously and the time reported in one project might actually origin from another. These circumstances had the consequence that even though some measures of efforts in past projects were existent, the usefulness of it is limited. Hence, to provide high accuracy, respondents considered expert estimates as the only reasonable option available.

III. Requirements on an Estimation Model

The investigated organization has a number of explicit and implicit requirements on the characteristics of an estimation model. To support the effort estimation performed in the project bidding and planning it has to comply with the process in which estimates are developed, and consequently fulfill a number of requirements. These requirements are described below together with a brief overview of the theory and research within these areas.

A. Requirements on a Estimation Model of the Implementation Projects

To provide sufficient input to the cost estimates developed during the bidding, the effort estimates need to be accounted for with respect to the activity and function they origin from. The same requirement for detail level goes for the effort estimates used in project planning. Hence, even though effort could be distributed top-down from an estimate of the project as a whole, estimates of different activities and system parts is necessary to enable costs to be calculated.

The initial estimates need to be developed quickly and is often based on uncertain assumptions initially to be successively refined and adjusted as more information is gained. Hence, to provide support, a model has to be capable of quickly producing rough initial estimates and provide means for these to be updated when additional information is provided about the project during bidding or planning. The desired model should provide experience based estimates that are usable as support and supplement in discussions among regarding the effort in projects. Therefore the estimates need to be supplied in a manner which is conceptually aligned with the organizations view of projects. Moreover, it has to be clear how the estimates are derived and what assumptions they are based on to enable them to be analyzed and possibly adjusted.

The effort required in projects naturally depends on numerous different circumstances with various impacts on effort. Notably is that all of these are not assessable in the early phases and some are only relevant to a very limited number of projects. The respondents found a couple of factors which was particularly important and ought to be considered when estimates are developed to provide a decent level of accuracy.

The inherent uncertainty of estimates were traditionally handled separately from the effort estimates and where addressed on a more general level. Although there was an apparent uncertainty in estimates, especially the early ones, there was some skepticism about involving uncertainty directly into the estimates. As a respondent from the sales department expressed it regarding potential use of uncertainty measures together with the estimates: "*I do not know how it would be used. People use intuition when assessing risks anyway and I don not think we need some kind of scientific calculation of confidence intervals or so.*". Other respondents did however acknowledge the need for uncertainty indications in estimates. They stated that uncertainty indications would provide guidance to estimators in identifying potential weak-spots in an estimate, and thereby provide support in the refinement process. Indications of uncertainty in the estimates were however only viewed upon as a tool during the estimation process and not an end in itself. The goal was to find the most accurate estimate in the end, and not the uncertainty of the estimate.

Summarizing the above, the following features were desired from an estimation model to provide support during the effort estimation made in early phases:

- Produce estimates that are detailed enough to perform cost calculations and to use in planning
- Consider the most important factors regarding effort and be possible to assess in the early phases.
- Provide estimates that are transparent, and make the output traceable and understandable.
- Provide some indication of uncertainty in estimates.

The basic idea behind these requirements was that the model should provide an initial estimate which was stepwise refined to better capture the conditions of the project at hand. The possible configurations of factors influencing project scope and effort is virtually infinite and it would be tricky to incorporate all this knowledge in an estimation model. Especially since existing data on past achievements was not suitable as input to the model, and it consequently had to be based on expert estimates. Hence, it would be sufficient if the model incorporated the most significant effort drivers. The less significant factors could be dealt with based on experience and judgment when this was required and necessary information were available.

B. Support of Requirements in Literature

Effort estimates are used both in the bidding of project and in the planning performed when a project is won. As discussed in Grimstad and Jørgensen [9], the exact meaning of the term effort estimate is somewhat unclear and planed effort, price-to-win and estimates of likely outcome is often mixed in the software effort estimation field. In fact, even if price-to win and planned effort are set aside there seem to be some differences in what an effort estimate should represent. Some (e.g. the PMBOK guide [20]) state that the estimates of cost and duration should represent the likely outcome while others claim that the natural output of any estimation process is the expected outcome [5]. These two values are not the same as the most likely effort when probability distributions are skewed. Most likely effort is the highest point in a probability distribution; expected effort is the mean.

Regardless if estimates should represent the most likely outcome or the expected outcome, there seem to be a strong support for assessing the uncertainty in estimates (see for example

[10][20][14][5][11]). Information about the uncertainty in estimates for example provides means for effective scheduling [6], project risk management and project portfolio management [15]. Based on this, the desire of uncertainty indications in the estimates has a strong support in research. Moreover, since models themselves are a source of uncertainty [15] the demand of a transparent, understandable and traceable model is healthy since it enables estimators to assess this uncertainty and possibly deal with it.

Regarding accuracy, some comparisons of approaches and models for effort estimation of software projects can be found in [7][12][17][24][4][11]. It is often suggested that different approaches and models should be used together to achieve an accuracy as high as possible [1][21]. Perhaps the most appropriate comparison in this context is made by Stensrud [26], who evaluates the applicability and accuracy of software estimation models in effort estimation of ERP projects. The conclusion from this evaluation was that the only software development approach that makes real sense in the context of pre-developed software was regression analysis, which in turn requires data to analyze.

Another approach that is not as dependent on data of past achievements is expert judgment. Expert judgment is however often frowned upon due to its subjective nature and has some of the drawbacks in comparison to models, such as its black box nature, the difficulties to reuse and modify the estimates, prone to various types of bias and the dependence on experience [22]. Despite these weaknesses, there is no strong evidence of models outperforming expert judgment in software effort estimation [11], and the use of models together with models is both common and regarded as good practice [22][10]. Based on this, the wish for a model to support the estimation process rather than replacing it is not controversial. Stensrud and Myrtveit [27] do for example state that models should be evaluated with respect to how they improve the performance of the humans making the estimates.

The requirements presented in this study are based on the idea that the model is supposed to be a tool. It is supposed to aid the estimators during the estimation process by gathering relevant experience and provides this in a traceable and transparent format. By doing this it limits many of the weaknesses involving expert estimates such as the dependency of experts being available to provide estimates; the black box nature that is inhibiting modification; and reuse of estimates. Furthermore, providing the estimators with an experience based starting point could help estimators to separate estimates of the most likely or expected effort from the bidding and planning. This separation is desirable since the three matters have conflicting goals [11].

Table 1. Requirements from the case study and requirements from previous research.

	Requirement found in the case study	Requirements found in literature
Detail Level	Produces estimates that are detailed enough to perform cost calculations and to use in planning	The bottom-up estimate, also referred to as the grassroots buildup estimate, is generally considered the most accurate one [13][18].
Model Comprehensiveness	Should consider the most important factors regarding effort and possible to use in the early phases.	According to Kitchenham and Linkman [15] models should include all relevant factors. If a factor is unknown an assumption should be made and risk should be adjusted.
Transparency/ Traceability	Provides estimates that is transparent, and makes the output traceable and understandable.	Jørgensen suggest that only models that are fully understood should be used [10] and Stensrud [27] propose that models should be evaluated on how well they provide support to humans developing the estimates.
Uncertainty	Provide some indication of uncertainty in estimates.	The support for assessing the uncertainty is strong, see for example [10][20][14][5][11].

IV. The Proposed Model

The model proposed here is developed in collaboration with the investigated enterprise to fit the circumstances surrounding their projects and to meet the requirements of this organization. Although it is made with the objective of suiting the investigated organization, it is most likely meet the requirements of organizations which deliver pre-developed software in a similar manner. The design of the model to meet the requirements are described below, as well as a description of its implementation in this case study.

A. Model Rationale

When selecting an estimation approach to suit the projects, a number of issues were considered. Firstly, there is an obvious need of traceability and detail level in the estimates to support the bidding and planning. Secondly, the scope of project is fairly predictable while at the same time quite variant. Three variables determine the scope of a project: the delivery model, the function scope and the activities which are performed by other parties. While the possible combinations of these three variables produces numerous different possible scopes, the content of a certain scope is easy to define and straightforward to assess. Thirdly, the effort required within a certain scope and the uncertainty of the predictions is in large dependent on handful of factors, such as the quantity and quality of network data dealt with in an implementation. These have quite big impact on the effort and the uncertainty of an estimate and even if the impact varies among functions and activities, the relation between a variable and the effort is known and possible to quantify. Fourthly, there are no reliable historic data available that would serve well as a basis for an estimation model. Fifth and finally, the model has to enable an initial estimate quickly and enable this to analyzed and adjusted as more information is gained.

The need for detail about the origin of estimates eliminates the possibility of using approaches producing estimates of whole projects only. Furthermore, the persons involved in the projects are specialized on specific domains and none of them would be able to accurately estimate effort in whole projects, something that would further complicate approaches taking an top-down perspective. A bottom-up estimate of the projects would hold all the desired detail level and transparency, and it is generally regarded as the most accurate estimation approach (see for example [13][18]). In general there are two reasons for not choosing the bottom-up approach when effort is estimated: it requires a clear definition of scope, and it is time consuming [18]. Both of these are however possible to overcome in this context if the bottom up approach is combined with the use of parameters, since:

- The work required to carry through a project is well understood as long as the functions included, the delivery model and the activities to be performed are known.
- The modular architecture of the system makes it possible to model changes in function scope by simply adding or removing functions and the activities associated with it.
- The impacts of variables are known on lower levels in the work breakdown structure.

Hence, the reasons for not choosing the detailed and accurate bottom up approach should not be considered an obstacle in this case. The requirement of clearly defined scope is not an issue since the content of projects is well known and their scope is easy to assess as long as the delivery model, the functions included and the activities performed is known. The tedious work it would be to estimate the effort of each and every combination of variables that can occur, is partially solved by combing functions and activities that can be estimated independently; partially solved by the use of complexity variables with discrete levels. The impact of these complexity variables are known on work package level and can be modeled as independent factors. Remaining is the different delivery models which due to the absence of distinct relationships between them have to be treated as two separate cases and modeled separately. The number of estimates needed when modeling the effort like this is however possible to overcome.

B. Model Design

The models available to estimate software project effort is either based on analogous models or parametric [8][13][18]. Analogical models, such as ANGEL [25] are based on a database of previously performed projects. The estimates are usually derived from this database by quantitative methods where the project at hand is compared to the ones in the database to find a similar one. This naturally requires a database of significant size. The parametric approach attempts to extract general patterns from previous projects and use these patterns to develop estimates of future undertakings. The parameters used to estimate effort in these models are usually lines of code or function points, measurements that seem inappropriate when estimating the effort of implementing pre-developed software [8][26].

In traditional models for software cost estimation the main variables are the size of the project and the complexity of it [8]. The estimation model proposed in this paper is similar to these models in the sense that it uses size and complexity to estimate the required effort. The scope of a project is in this case used as a measure of the size; the other variables influencing effort (e.g. data quantity) can be considered as variable influencing the complexity of projects.

The proposed approach is based on a decomposition of projects into manageable and independent parts. Experts are the giving estimates on the various parts of projects and on how a set of variables influence this effort. Compared to traditional approaches, which rarely produce estimates of other than whole projects, this is beneficial since it enables stakeholders to review the estimates in a transparent and readily expressed format. The produced estimates thereby provide a ground for discussions about their size and accuracy, something which could be difficult if only a formula and a total estimate where produced. Another significant difference from traditional approaches is the feasibility to base it on expert estimates, rather than vast datasets of previous projects.

To attain an estimate of a project, the scoping variables are used to select the work packages included in the scope. Given the values of the complexity variables effort estimates of the effort for individual work packages can be derived (cf. Fig. 1). The effort of work packages is derived by multiplying a nominal expected effort with a factors corresponding to the degree of complexity in the project. The effort of work packages that are included in the scope are aggregated to higher levels (a Function Package is zoomed in Fig. 1).

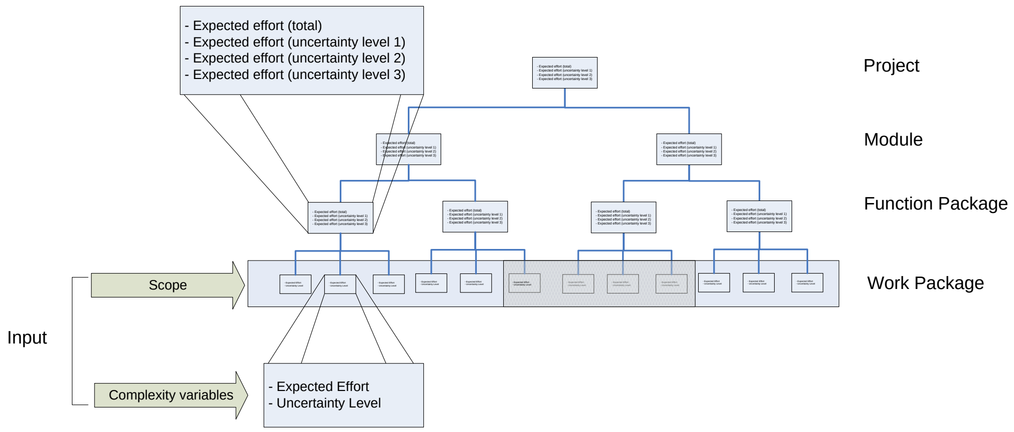


Figure 1. Model overview. The scope determines the work packages included and together with complexity variables produce effort estimates of work packages that are aggregated. In the figure, one work package and one function package is focused.

Since the estimates need to be additive to allow aggregation as describe above, the expected effort need to be estimated. Estimates of most likely effort are not additive unless the probability distribution is symmetric [5], and if they are symmetric they hold the same value as the expected effort. A common and straightforward way of expressing uncertainty in estimates is the use of prediction intervals; this is also used in this suggested model. However, to simplify the model and limit the assumptions it is based on, the uncertainty is expressed as one of a number of predetermined intervals. For example: -60 to +150 percent, -30 to +50 percent and -5 to +10 percent of expected effort. This approach is chosen since prediction intervals are not as simple as the expected values to combine and aggregate, especially if the intervals covariate some extent. Instead of making extensive and questionable assumptions such as the ones needed to combine them mathematically by using PERT-formulas or similar [29], it is visualized on each level in the work breakdown structure how much time and how many activities beneath that are of different levels of uncertainty. Hence, estimates do on work package level comprise of an estimate of expected effort and a level of uncertainty, corresponding to a predetermined interval. On higher levels in the work breakdown expected efforts are gained by additions; and uncertainty is visualized by showing how much effort and how many activities there are of each uncertainty level. For example, 10 weeks and 4 activities of uncertainty level 1, 4 weeks and 3 activities of uncertainty level 2, and so on.

One could, as often is done, assume the nature behind the distribution and by means of mathematics combine them into a single interval. There are however apparent dependencies among the activities of projects which could require the covariance between different efforts to be quantified and considered in the calculation. Omitting this covariance and assume that the efforts are statistical independent would be a false assumption. Since the covariance between variables would have to be expressed to calculate the true prediction intervals, such an analysis would be extremely difficult to perform when exhaustive data on projects is absent. The approach with different levels of uncertainty, which is not merged into a single interval, provides the necessary indicators of uncertainty. Anyone who wish to further analyze the uncertainty from different perspectives will by this be aided in identifying uncertain estimates, and this was also the desired from the organization investigated.

To enable a clear path between the collected data and how this data is interpreted when the model is used the levels in complexity variables are expressed by a number of discrete levels. Levels could for example relate to a specific number (e.g. 35) of objects that need to be included in the power network model. The use of a limited number of discrete grades has some obvious shortcomings since it limits the scope of the model. It is however motivated by the difficulties in finding some kind of continuous functions for each complexity variable and its impact. This would require assumption to be made about the relations and would not necessarily yield better estimates. Furthermore, it is possible to assess combinations of these discrete values using for example weights, fuzziness or interpolation between the values.

The data required by the model is firstly the effort and uncertainty is estimated in a nominal state of the complexity variables, i.e. a specific state of each variable. Based on this the variables deviation from nominal state can impact both the expected effort and the uncertainty of it. The plausible impact on expected effort is modeled as a factor which is multiplied on the effort and the plausible impact on uncertainty is modeled by increasing or decreasing the uncertainty level.

This model is like all models based on assumptions about the reality. Firstly, the model assumes that impact of parameters could be modeled as factors which impact work independently of other parameters impact. This assumption implies that the impact of one variable is unaffected by the state of other variables. Secondly, assumptions are made on the work the projects entail. The work entailed in a project is assumed to be driven by the functions that it contains, and it is assumed that a list of functions and activities together with one of the delivery models is a good reflection of the scope of a project. Thirdly, the model assumes that estimates of work on system parts can be made independently of which other parts that is included. This is possible since the system is modularized and the functions that are having dependencies between them in terms of effort are also functionally dependent of each other. Hence, the constraint in terms of possible combinations of functions this causes is not an issue in using the model, even though there are dependencies among them.

C. Collected Estimates and Tendencies in These

The model is based on a breakdown of the work performed in projects and the number of estimates needed is dependent on the size of this work breakdown and the number of impacts from complexity variables. For the model implemented in this case study the number of estimates needed exceeded one thousand, including estimates of complexity variables impact. To develop the estimates the wideband Delphi procedure [2] was used and nine persons with expertise in different domains of the system supplied estimates for their particular domain. Estimators were instructed to estimate the expected effort together with the uncertainty of this prediction in one of four uncertainty levels, ranging from within -10/+20 percent to outside of -50/+100 percent of the expected effort with 90 percent confidence. In this uncertainty they were instructed to include both variability and ambiguity, as described by Chapman and Ward [5].

The model was implemented in a spreadsheet and enables estimates to be visualized in various ways on different levels in the work breakdown. As it is implemented, it also allows estimates to be analyzed and adjusted if this is appropriate judging from the circumstances of the project. The model implemented in this case study uses a handful complexity variables that according to respondents capture project complexity in large. On the lowest level, estimates are given by the expected effort and the estimate uncertainty level. On aggregated levels, it is possible to view how many man hours that is expected and which uncertainty levels the estimates underneath hold (cf. Fig. 2). This allows estimators to get an instant grasp of the underlying estimates uncertainty and trace these down to individual work packages, making identifications of potential weak spots simple and straightforward. Furthermore, it renders the overall uncertainty of the project and different parts of it in a quick and straightforward way. Since the number of activities holding different uncertainty levels could be of relevance for the total uncertainty, this can be visualized as well.

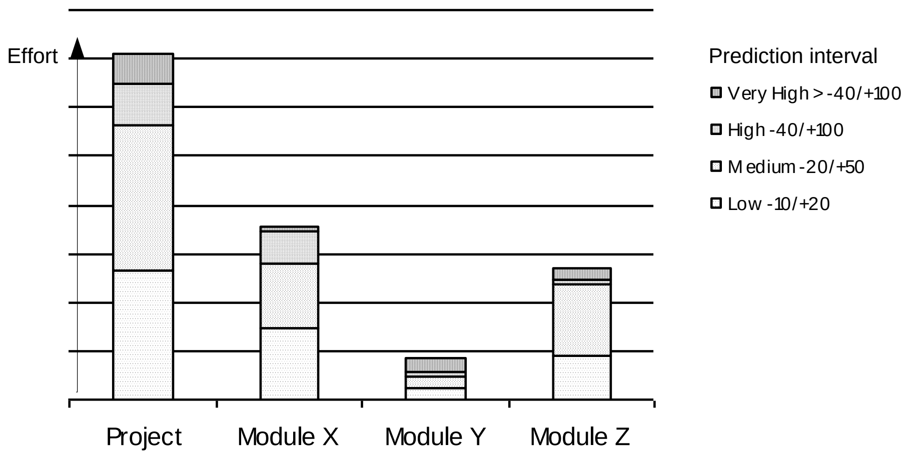
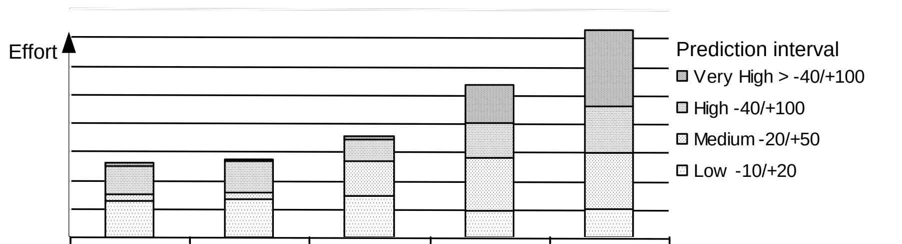


Figure 2. Effort visualized on different levels in the project decomposition.

When altering one complexity variable from the nominal state the estimates produced by the model will change in accordance with the circumstances. Circumstances which lower the effort and uncertainty will naturally lower the uncertainty and expected effort of the model, and vice versa. Altering a complexity variable (e.g. the network size) will have an impact on numerous work packages and their corresponding system modules. The design of the model utilizes impact of the variables as independent factors, and combines them by treating their impacts one by one, independent of how other variables affect the estimates. The consequence of this is that if two variables have an impact on the same element in the work breakdown structure, their impacts on expected effort will be multiplied, and their impact on uncertainty will be added to each other. This makes the momentum of two or more variables with similar impacts stronger than the individual impacts of each of the variables. Fig. 3 shows an illustrative view of how changes in complexity variables impact the estimates within a specific scope.



Low	Low	Nominal	Nominal	High	Complexity Variable 1
Low	Low	Nominal	High	High	Complexity Variable 2
Low	Nominal	Nominal	High	High	Complexity Variable 3

Figure 3. Example of estimates for different values in complexity variables within a scope.

Although the estimates shown here are fictive, they illustrate the tendencies in the collected estimates well. As shown here is the span between the smallest effort and the biggest effort quite large, and the difference in certainty about the effort needed in various projects is likewise significant. In the estimates collected in the case study, the extreme combination of complexity variables yielded estimates that to about a quarter held an uncertainty of the highest uncertainty level, an interval reaching somewhere outside of -40 to +100 percent of the expected effort. Around a quarter of the estimated hours were on an interval outside of -20 to + 50 percent and only about an eight of the estimates held the lowest uncertainty in this case.

V. Summary and Conclusions

This paper has described a case study performed at a worldwide engineering company executing delivery projects of pre developed software. The case study investigated the needs of effort estimation support and based on this a number of requirements was developed on an effort estimation model to be applicable in this context. In short the requirements surfaced from the case study address the need of quickly being able to develop estimates and providing these in a transparent and traceable way. High accuracy and a wide model scope are naturally desirable, however, the respondents are well aware of the great variations that occur in scope and complexity; and find it very unlikely that a model would be able to capture all this. Instead they wish for a model that provides an experienced based first estimate that can be used as starting point and possibly adjust this to correspond to the circumstances of the project at hand as more detailed information is gained. Furthermore, since no reliable data on past achievements where available, the model should be based on expert judgment.

There are several tools commercially available which produce effort estimates by statistical relationships, algorithms, or similar means. The parametric bottom-up approach proposed in this study is far simpler than most of these, and naturally this is one of its strengths. Its ability to keep the model simple and still incorporate the important factors when estimating effort makes it easily applied and at the same time powerful enough for the situation at hand. The effort can be visualized on various levels and from different angles to ease analysis and refinement. Moreover, the visualization of uncertainty in the estimates enables estimators to both get an instant grip of the uncertainty behind the estimate and provides means for this uncertainty to be traced to specific components and activities.

This study confirms what previous studies have found. That there is a difference in estimating effort needed to implement pre-developed software and to estimate the effort of development activities [8][26]. The model proposed in this paper assumes that scope is known early on and can be decomposed in a way that is applicable to all potential projects. This is usually not the case in software development projects and therefore its applicability to such projects is limited. Something that could be exemplified by the opinion among the individuals performing the estimation within this study, who see the part of projects involving standard functions as the ones which is easy to predict. It gets difficult first when deviations from standard exist and software development is required. As a model for estimating effort of projects implementing pre-developed software it is however applicable. The type of implementation projects performed by this organization is not very different from projects implementing pre-developed software in general. The requirements made on an estimation model in this study and the proposed solutions are accordingly likely to be applicable in other organization and for other systems. For example projects implementing ERP solutions.

References

[1] Armstrong, J,.S.; Principles of forecasting – A Handbook for Researchers and Practitioners, Kluwer Academic Publishers Group, Netherlands, 2001.

[2] Boehm, B.W, “Software engineering economics”, Prentice Hall, NJ, 1981.

[3] Boehm, B.W, Horowitz, E., Westland, C. Madachy, R.; Selby, R.; “Cost models for future software processes: COCOMO 2.0”, Annals of Software Engineering., vol. 1, no. 1, pp. 57-94., 1995-

[4] Briand, L.C., Emam, K.E., Surmann, D., Wieczorek, I., Maxwell, K.D.; “An assessment and comparison of common software cost estimation modeling techniques”, International Conference on Software Engineering, Los Angeles, USA. ACM, New York, pp. 13–323, 1999.

[5] Chapman, C., Ward S.; “Managing project risk and uncertainty: A constructively simple approach to decision making”, Chichester, UK, John Wiley & Sons, 2002.

[6] Dawson, R.J., Dawson, C.W.; “Practical proposals for managing uncertainty and risk in project planning”, International Journal of Project Management vol. 16, no. 5, pp. 299-310, 1998.

[7] Finnie, G.R., Wittig, G.E., Desharnais, J.M.; “A comparison of software estimation techniques: using function points with neural networks, case-based reasoning and regression models”, Journal of Systems and Software, vol. 39, issue 3, 281–289, 1997.

[8] Francalanci, C.; "Predicting the implementation effort of ERP projects: Empirical evidence on SAP/R3", Journal of Information Technology, vol. 16, no. 1, 33-48, 2001.

[9] Grimstad, S., Jørgensen, M., Moløkken-Østvold, K.; “Software effort estimation terminology: The Tower of Babel,” Information and Software Technology 48, pp. 302-310, 2005.

[10] Jørgensen, M., Moløkken-Østvold, K.; “A preliminary checklist for software cost management”, Proceeding of International Conference on Quality Software, pp. 134-140, 2003.

[11] Jørgensen, M., “A review of studies on expert estimation of software development effort”, Journal of Systems and Software. vol. 70, no. 1-2, pp. 37-60. 2004.

[12] Kemerer, C.F.; “An empirical validation of software cost estimation models”, Communications of the ACM, vol. 30, no. 5 , pp. 416-429, 1987.

[13] Kerzner, H.; “Project management: a systems approach to planning, scheduling, and controlling”, John Wiley and Sons, New Jersey, 2001.

[14] Kitchenham, B., Pickard, L.M., Linkman, S., Jones, P.W.; “Modeling software bidding risks”, IEEE Transactions on Software Engineering”, vol. 29, no. 6, 2003.

[15] Kitchenham, B., Linkman, S.; “Estimates, uncertainty, and risk”, IEEE Software, vol. 14, no. 3, pp. 69-74., 1997.

[16] MacDonell, S., Gray, A, “Alternatives to regression models for estimating software projects”, Proceedings of the IFPUG Fall Conference, Dallas TX, IFPUG 279.1-279.15, 1996.

[17] Mendes, E., Watson, I., Triggs, C., Mosley, N., Counsell, S.; “A comparison of development effort estimation techniques for web hypermedia applications”, Proceedíngs of the IEEE Metrics Symposium, pp. 131-140, 2002.

[18] Milosevic, D.; “Project management toolbox”, John Wiley & Sons, New Jersey, USA, 2003.

[19] Molokken, K., Jørgensen, M.; “A review of software surveys on software effort estimation”, Proceedings 2003 International Symposium on Empirical Software Engineering, ISESE, pp. 223- 230, 2003.

[20] Project Management Institute; “A guide to the project management body of knowledge (PMBOK® guide)”, version 3.0, Pennsylvania, USA, 2004.

[21] Pressman, R.S.; “Software engineering: a practitioner’s approach”, Sixth international ed, McGraw-Hill, NT, USA, 2005.

[22] Roy, R.; “Cost engineering: why, what and how?”, Decision Engineering Report Series, Cranfield University, 2003.

[23] Schofield, C.; “Non-algorithmic effort estimation techniques”, ESERG: TR98-01, Bournemouth University, Department of Computing, 1998.

[24] Shepperd, M.., Schofield, C.; “Estimating software project effort using analogies”, IEEE Transactions on Software Engineering, vol. 23, no. 12, 1997.

[25] Shepperd, M., Schofield, C, Kitchenham, B.; “Effort estimation using analogy”, Proceedings of the 18 th International Conference on Software Engineering, pp. 170-178, 1996.

[26] Stensrud, E.; “Alternative Approaches to effort estimation of ERP projects”, Information and Software Technology, vol. 43, pp.413-423, 2001.

[27] Stensrud, E., Myrtveit, I.; ªHuman performance estimating with analogy and regression models: An empirical validation”, Proceedings of. Software Metrics Symposium, METRICS'98, pp. 205-213, 1998.

[28] Walkerden, F., Jeffery, R.; "An empirical study of analogy-based software effort estimation, empirical software engineering", Volume 4, Number 2, pp. 135-158, 1999.

[29] Williams, T.; "Modeling complex projects", John Wiley & Sons, West Sussex, United Kingdom, 1998.