Profitability Estimation of Software Projects: A Combined Framework

Stefan Wagner, Songmin Xie
Institut für Informatik
Technische Universität München
Boltzmannstr. 3
85748 Garching b. München, Germany
{wagnerst,xies}@in.tum.de

Matthias Rübel-Otterbach, Burkhard Sell
Kabel Deutschland Breitbandservices GmbH
Betastr. 6–8
85774 Unterföhring, Germany
{Matthias.Ruebel,Burkhard.Sell}@kabeldeutschland.de

Abstract

Decisions on carrying out software projects are a recurring problem for managers. These decisions should ideally be based on solid estimates of the profitability of the projects. However, no single solution has been established for this task. This paper combines the German WiBe *framework for costs and benefits of IT projects with certain cost estimation approaches in order to ensure reliable profitability estimates. The applicability of the framework is shown in an industrial case study,*

1. Introduction

A central and recurring question for software managers is: Should this project be carried out? This question is usually followed by: Is it profitable? Hence, methods for estimating the profitability of software projects should be in the toolbox of every manager in the software business. However, there is no established set of methods that are recommended to be used. Even for the cost estimation part alone. There is a large variety of possible methods to be used. Currently, expert estimation is the most commonly used technique [5]. However, it is not clear whether it is the most accurate or effective. Jørgensen showed in his review [5] that the 15 available studies on different methods are not conclusive. There are 5 studies that show that expert estimation is more accurate, 5 found no difference, and 5 found model-based estimation to be more accurate. For profitability, it is even difficult to find a fully-fledged method.

Problem. There is a lack in established methods for profitability estimation although this is a common, day-to-day problem in software management. Typically, managers resort to expert estimations. However, the empirical research has not been able to conclusively show that this method is most accurate.

Contribution. We propose a method for profitability estimation for software projects called *SW-WiBe* that is based on (1) a proven framework for IT profitability estimation and (2) the results of empirical research on software cost estimation. In essence, we employ the WiBe framework that has been in use for 15 years and that provides a means for estimating non-quantifiable benefits. The cost side is estimated by (at least) 3 different methods containing expert as well as model-based methods. To improve the accuracy, these methods are combined by the Wideband Delphi process. The applicability of *SW-WiBe* is shown in an industrial case study.

2. Profitability analysis

Profitability analysis is concerned with the relation of costs and benefits. Hence, it shows whether an endeavour is profitable.

2.1. Costs and Profitability

A large part of research in software economics deals with the estimation (or prediction) of the costs of software development and maintenance, e.g. [2, 4]. Although even in that area no conclusive results have been reached what approaches (expert or model-based) are better in what cases, this is only half the way. It is equally important to analyse the benefits, quantifiable and non-quantifiable, in order to decide on its profitability. This part is largely underdeveloped in software economics research [3].

2.2. WiBe

WiBe [8] stands for *Wirtschaftlichkeitsbetrachtung* and is a method for estimating and calculating the profitability of IT projects. It was developed for the German Federal Ministry of the Interior and has been improved several times over the last 15 years. It has been used in various public projects involving information technology. Hence, it is an established and proven method for profitability analysis of such projects with an emphasis on in-house development.

WiBe is suitable for a comprehensive analysis of software projects. It is especially interesting that it consists of a set of building blocks for the analysis and mainly proposes a framework. It does not prescribe specific methods for estimating costs and benefits. However, it considers non-quantifiable benefits explicitly and describes a utility analysis for them. In this way, the non-quantifiable benefits can be appropriately dealt with. The main modules of WiBe are the following:

- Monetarily quantifiable costs and benefits (WiBe KN)
  - Monetarily quantifiable costs
  - Monetarily quantifiable benefits
- Non-quantifiable benefits
  - Urgency (WiBe D)
  - Qualitative and strategic importance (WiBe Q)
  - External effects (WiBe E)

The monetarily quantifiable costs and benefits must be provided by some other estimation method. Then the net present value is calculated in order to account for their temporal distribution. The non-quantifiable benefits are handled using utility analysis, a standard approach for qualitative factors. In essence, experts assign points to various qualitative issues and these points form the basis for the decision on the profitability.

3. Combined Framework: SW-WiBe

The WiBe method provides a solid ground for analysing the profitability of IT projects. However, it mainly provides a framework for this analysis. Hence, we propose a concrete instantiation for software projects: *SW-WiBe*. We aim to fulfil two goals with our method:

1. Providing a complete method for the analysis of the profitability of software projects
2. Ensuring reliable estimates

To reach the first goal, we choose concrete cost estimation methods for software to be used inside WiBe. The second goal is supported by using several diverse methods and combining them based on the Delphi method [2, 10]. An overview of *SW-WiBe* is given in Fig. 1. This strategy is supported by Jørgensen [5] who showed that it is most beneficial to combine estimates from different experts and estimation strategies as well as to ask the estimators to justify and criticise their estimates.

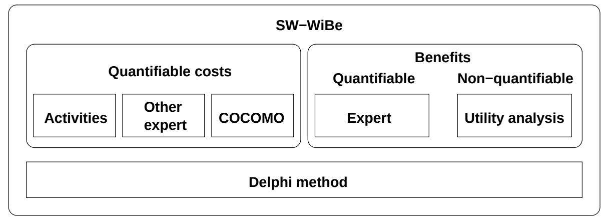

Figure 1. Overview of the SW-WiBe method

3.1. Quantifiable costs

As shown in Fig. 1, we propose to use (at least) three different methods for estimating the quantifiable costs of the project. In order to achieve reliable estimates, these methods should be diverse. Therefore, the activity-based method (an expert method) and COCOMO II (a model-based method) are an integral part of SW-WiBe. The third method can be flexibly fitted to the available competence. To improve diversity this usually is a different kind of expert estimation. This way, we include at least three different views on the estimation problem:

1. An activity-based expert estimation that brings in the experience of an expert. Furthermore, the structuring with activities allows an easy check of the estimates during the project.
2. A different expert estimation that introduces a different view and uses different experiences. It may use a different work breakdown structure (WBS) [10].
3. The model-based method COCOMO II that is based on an explicit model and empirical data.

For economically correct handling of the quantifiable costs, the distribution of the costs over time must be considered. For this, the standard method of *net present value* is available. It allows to calculate the present value of the whole costs by discounting them w.r.t. the point in time when they occur.

3.2. Quantifiable benefits

The estimation of the benefits of a software project is more difficult. WiBe structures this in estimating the quantifiable benefits and the non-quantifiable benefits. How non-quantifiable benefits are estimated is explained later.

However, for the quantifiable benefits, there are no common methods. Furthermore, Boehm and Sullivan [3] point out that "effective methods for modelling software benefits tend to be highly domain-specific." Hence, we cannot use a "one-size-fits-all" in this case. In many instances, an expert estimation based on available accounting data will be a possibility. In any case, the net present value of the estimated quantifiable benefits has to be used.

3.3. Non-quantifiable benefits

The most difficult part to handle are the non-quantifiable benefits. The standard WiBe suggests to use *utility analysis*. In essence, this is a ranking of the various influential criteria on a qualitative basis. This ranking uses points that are associated with the different qualitative ranks. An example for the criteria *stability of the legacy system: downtime* is shown in Tab. 1. It gives 6 possible qualitative ratings for the downtime of the system to be replaced and the corresponding points.

The influential criteria have been compiled based on the experience with IT projects from the original WiBe authors. However, this list can be tailored to the specifics of the project. The general classification is in (1) urgency, (2) qualitative and strategic importance, and (3) external effects as described in Sec. 2.2. Examples are *abidance by the law*, *reuse of existing technology*, or *acceleration of work processes*. A complete list can be found in [8]. Each criteria has a weight that reflects its importance and that is multiplied with the point value. Then all the weighted points are added for each of the 3 above mentioned classes of criteria. These sums are later used for decision-making.

3.4. Roles and workflow

All these results for the 3 WiBe parts are compared and adjusted in a Delphi process. The detailed workflow is depicted in the activity diagram in Fig. 2. There are 4 roles necessary for the application of SW-WiBe. First, the *Project manager* is supposed to have all the basic information about the project. For example, this information should contain specifications of the functional and quality requirements of the system to be built. Second, the *Expert A* uses an activity-based method to estimate the costs. This means that the project is broken down into activities that need to be performed to develop the system based on its specification. These activities are estimated separately and then combined to an overall estimation.

Third, *Expert B* uses a different estimation method than *Expert A*. This can be another kind of activity-based estimation or an estimation based on another WBS such as components of the software system. Fourth, the *COCOMO expert* uses the project information to estimate the size of the system and all the necessary parameters to perform a cost estimation. All these roles can be filled with more than 1 person and 1 person can work in more than 1 role. However, we suggest that at least 3 different experts work in the estimation process to ensure enough diversity. Furthermore, it is beneficial to have technical as well as non-technical professionals in these different roles. As Moløkken-Østvold and Jørgensen show in [7], professionals in technical roles tend to be too optimistic in their estimates. Hence, a combination can mediate this.

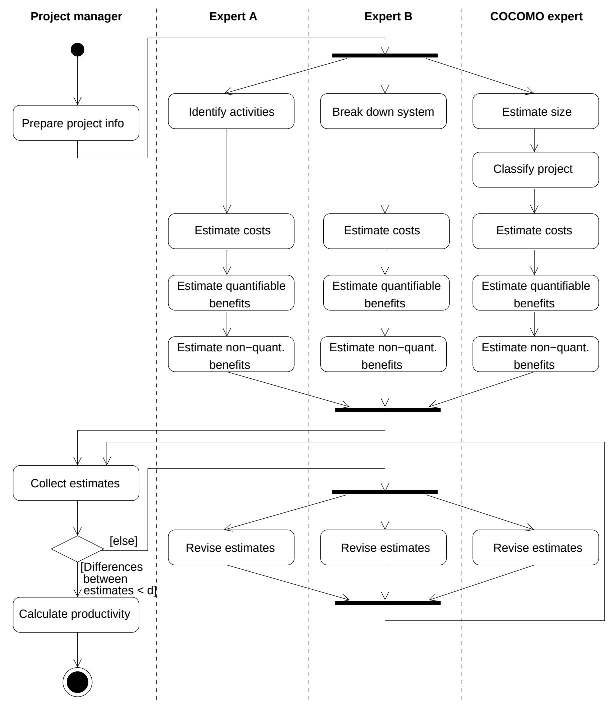

Figure 2. Activity diagram of the workflow

3.5. Delphi process

All 3 experts also give estimates for the quantifiable and non-quantifiable benefits of the software system. Although, they cannot use their diverse methods for cost estimation, the differing estimates are still useful for the Delphi process. This process is now used to adjust the estimates for the three parts. We use the *wideband delphi* as described in [2, 10] to adjust the estimates and minimise the variations. For this, the experts explain their estimates in a group session and are allowed to adjust them. These new estimates are again collected and discussed in another meeting. This is repeated until the variation lies below a threshold value d. This is usually a value of 10–15%.

To use such a group process is vital for reliable estimates. As Moløkken-Østvold and Jørgensen investigated in [6], es-

Table 1. Example criteria for *stability of the legacy system: downtime*

Points	0	2	4	6	8	10
Rating	not at risk	hardly affected	tolerable	troublesome	highly troublesome	intolerable

timates based on a group process have higher accuracy as the single estimates or even as the direct average of several individual experts. The latter is also described by Shepperd in [9]. Moreover, He proposes the combination of several predictors as one of the main research challenges.

3.6. Decision

For the profitability estimate, we use the 4 modules of WiBe as explained in Sec. 2.2: quantifiable costs and benefits (KN), urgency (D), qualitative-strategic importance (Q) and external effects (E). The KN module is used to calculate the *basic profitability*. Taking the modules D, Q and E into account allows to estimate the *extended profitability*.

Basic profitability. The basic profitability is simply the quantifiable profitability. Hence, it constitutes the difference of the quantifiable benefits and the quantifiable costs. Depending on the context, this should usually be divided in the development costs and the maintenance costs. The costs as well as the benefits need to be distributed over the estimation period (usually 5–10 years). Based on the temporal distribution, the net present value (NPV) is calculated using standard methods. The basic profitability (BP) is then:

$$BP = \text{NPV}(\text{quantifiable benefits}) - \text{NPV}(\text{quantifiable costs}) \quad (1)$$

It can already be used for decision making. In case the BP is positive, i.e. the quantifiable benefits are greater than the costs, the project should definitely be carried out. If the result was a negative BP, the non-quantifiable criteria should be considered in the extended profitability.

Extended profitability. The extended profitability (EP) introduces the non-quantifiable aspects of urgency (D), qualitative-strategic importance (Q) and external effects (E) into the decision making. As described above, the weighted points for each module have to be accumulated. These sums represent the non-quantifiable necessity of the software project. There is a set of rules that guides in decision-making based on these points. They are summarised in Tab. 2.

The project *must* be carried out in case the current system does no abide by the law (any more). There is obviously no way to avoid this. It might, however, be possible to improve the BP by decreasing the costs. The second rule highly recommends to execute the project in spite of a negative BP in case it is central to the general IT concept of the company. Other future developments in the software landscape of the company may depend on this project and hence it can be necessary to build it. Finally, the standard rule is that the accumulated points of at least one module need to be higher than 50 in order to carry out the project. This implies that the project is either significantly urgent, strategically important or has significant external effects that justify the execution of the project. In the two cases in which the project *should* or *can* be carried out, obviously the amount of quantifiable costs needs to be strongly considered in relation to the benefits in order to come to a decision.

Table 2. Decision rules for the extended profitability

Guard	**Result**
Abidance by the law = not abided (10 points)	Must be carried out
Significance inside the IT concept = key position (10 points)	Should be carried out
$D > 50 \vee Q > 50 \vee E > 50$	Can be carried out

4. Case study

The applicability of the proposed method to a real industrial environment is shown in a case study with a large German cable network operator. The profitability of a web portal project is analysed.

4.1. Environment

Kabel Deutschland Breitbandservices GmbH is the leading cable network operator in Germany. The company provides TV, radio, Internet and telephony via its cable network. It employs about 2,500 people in seven locations. The department *Web Applications* is a service provider for the other departments by providing infrastructure for process support. The department develops the Internet and applications for the agents, marketing staff and end customers which are connected to the business logic, core applications and logistics of the company.

Over 3 years, 7 web portals have been developed that serve for the communication with these different stakeholders. Currently, it is foreseeable that new requirements will come up for these portals. Hence, the software consulting house *softlab* was entrusted with providing proposals for a restructuring of the portals. This proposal contains a unification of the different portals based on a uniform technology. The management at Kabel Deutschland Breitbandservices GmbH is now interested in the profitability of the proposal. The profitability analysis will be the basis for the decision on carrying out the project.

4.2. Profitability estimation

Roles. The roles of the SW-WiBe are filled with personnel as follows: The role of the *Project manager* is performed by the department manager at Kabel Deutschland Breitbandservices GmbH (one of the authors) using information from the study of softlab. He together with the second author from the company occupy also the role of *Expert A*. *Expert B* are the technology experts at softlab. The authors of TU München together with experts of Kabel Deutschland Breitbandservices GmbH fill the role of the *COCOMO expert*.

Costs. All experts made their estimates based on a specification of the necessary solution for the unified web portal. *Expert A* used the *percentage method* [1] in which one project phase (implementation) was estimated and extrapolated to the other phases based on experience data. *Expert B* used an activity-based method, i.e. breaking down the project into activities and estimating each activity individually. The *COCOMO expert* made a size estimate based on the existing portals and determined the necessary COCOMO parameters in expert interviews at Kabel Deutschland Breitbandservices GmbH.

The three individual estimates were then compared in the Delphi process. It turned out that *Expert A* and *Expert B* made rather close estimates whereas the COCOMO estimate was a magnitude higher. Hence, this result was reexamined and errors in the size estimate were uncovered. In the second estimation round, the estimates were inside the range of $d < 15\%$. The cost estimates are depicted in Fig. 3. The final agreement was to use the average of 1,678 person-days.

Benefits. It was decided by the *Project manager* that there are no benefits of the project that are currently quantifiable. Hence, the experts concentrated on the non-quantifiable benefits. Two experts, *Expert A* and *Expert B* performed a utility analysis on the basis of the criteria provided by WiBe (cf. Sec. 2.2). The results divided into the three modules of non-quantifiable benefits are provided in Tab. 3. The results were not subject to a further Delphi process because (1) they have been discussed inside Kabel Deutschland Breitbandservices GmbH and softlab separately already and (2) the

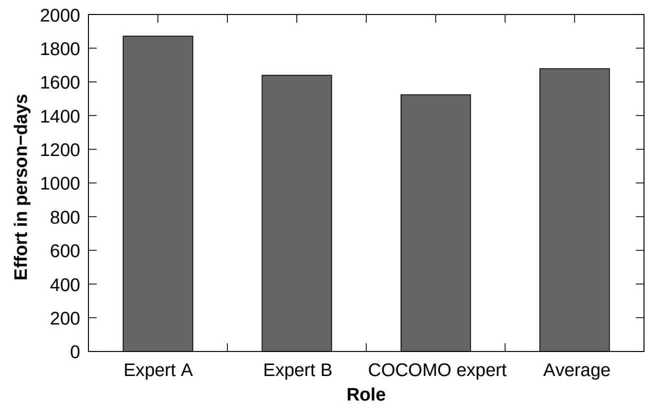

Figure 3. The cost estimates of the different roles

Project manager wanted to see the differences between the internal and external opinions.

Table 3. Utility analysis of the non-quantifiable benefits

Module	Expert A	Expert B
Urgency (D)	37	19
Qualitative/strategic (Q)	43	50
External effects (E)	44	52

Decision. The collected results for the costs and benefits need now to be combined in order to reach a decision about the project. The quantitative module of SW-WiBe can easily be calculated because there are no quantifiable benefits. Note that we expect all the costs to occur in the first year and hence no discounting is applied. Hence, the *basic profitability* for the project is negative:

$$\text{BP} = -1678 \text{ person-days} \approx -1,360,000 \text{ Euro} \quad (2)$$

Following our decision rules, the project should not be carried out unless the utility analysis can provide counterarguments. The first two rules from Tab. 2 do not apply for this project. Neither are any laws not abided by the legacy system nor has the project (currently) a key position in the IT concept of the company. These ratings were consistent for both experts.

Then the third rule describes whether at least one of the modules urgency, qualitative and strategic importance, or external effects is strong enough to justify the project. As can be seen in Tab. 3, none of the module ratings of *Expert A* are higher than 50 and only the external effects are rated as such from *Expert B*. The *Project manager* sees this as not enough justification to carry out the project. Nevertheless, the ratings are close to 50, especially for WiBE Q and WiBe E. This means that the necessity of the project might change in the future. Moreover, we have to note that it is planed to investigate possible quantitative benefits which might change this decision as well.

4.3. Discussion

The case study with Kabel Deutschland Breitbandservices GmbH demonstrates a real world application of the SW-WiBE framework. It shows that SW-WiBe is applicable to such situations and that it can provide guidance for the whole profitability analysis process. The combination of several cost estimation methods led to an estimate that is far more trustworthy than single estimates alone. This can be seen in the fact that we actually discovered an error in the application of COCOMO by comparing it to the other estimates in the Delphi process.

However, the cost estimate alone would have been difficult to use as a basis for the decision about carrying out the project. If the costs lay beyond the possible budget, the project could not be done anyway. If this is not the case, we will need further guidance. This guidance is given by the utility analysis of SW-WiBe. It ranks important non-quantifiable or difficult to quantify factors and combines them in three modules that affect the decision. This was perceived as very useful in the case study.

5. Conclusions

Estimating the profitability of a software project to be done is a common problem in practical software engineering. A method that helps in that estimation process would be a useful tool in the toolbox of software managers. However, there are only few such approaches.

We propose *SW-WiBe* as a method for such profitability estimations for software projects. It is based on the WiBe framework for profitability of IT projects. Based on current research results, the framework is filled with diverse expert and model-based cost estimation methods that are combined by a Delphi process. This improves the reliability and accuracy of the estimates. The difficult part of the non-quantifiable benefits is handled by utility analysis that provides a set of important criteria and possible rankings. The quantified estimates together with the utility analysis result in the decision about the project's profitability.

SW-WiBe was applied in a real industrial environment. A project for the restructuring of the web portals of a large German cable network operator was analysed. The method proved to be applicable in that environment. The combined cost estimates as well as the utility analysis was perceived as very useful for reaching the profitability decision.

We plan to apply the SW-WiBe in further case studies. Obviously, we need to go along with the planned projects (in case they are carried out) and to compare the estimates with the actual costs and benefits. This would allow us to test the hypothesis that the estimates are more accurate and reliable more formally.

Acknowledgements

We are grateful to the engineers from softlab for their help on the specification and their estimates.

References

[1] H. Balzert. *Lehrbuch der Software-Technik*. Spektrum Akademischer Verlag, 1996.

[2] B. W. Boehm. *Software Engineering Economics*. Prentice Hall, 1981.

[3] B. W. Boehm and K. J. Sullivan. Software economics: A roadmap. In *Proc. 22nd International Conference on Software Engineering (ICSE '00)*. ACM Press, 2000.

[4] L. C. Briand, K. El Emam, and B. F. Cobra: A hybrid method for software cost estimation, benchmarking, and risk assessment. In *Proc. International Conference on Software Engineering (ICSE '98)*, pages 390–399. IEEE Computer Society, 1998.

[5] M. Jørgensen. A review of studies on expert estimation of software development effort. *The Journal of Systems and Software*, 70:37–60, 2004.

[6] K. Moløkken-Østvold and M. Jørgensen. Group processes in software effort estimation. *Empirical Software Engineering*, 9:314–334, 2004.

[7] K. Moløkken-Østvold and M. Jørgensen. Expert estimation of web-development projects: Are software professionals in technical roles more optimistic than those in non-technical roles? *Empirical Software Engineering*, 10:7–29, 2005.

[8] P. Röthig, K. Bergmann, and C. Müller. WiBe 4.0. Empfehlung zur Durchführung von Wirtschaftlichkeitsbetrachtungen in der Bundesverwaltung, insbesondere beim Einsatz der IT. Schriftenreihe der KBSt, Band 68, Bundesministerium des Innern, 2004.

[9] M. Shepperd. Software project economics: a roadmap. In *2007 Future of Software Engineering (FOSE '07)*, pages 304–315. IEEE Computer Society, 2007.

[10] A. Stellman and J. Greene. *Applied Software Project Managment*. O'Reilly, 2005.