Science of Computer Programming 74 (2009) 900–933

Contents lists available at ScienceDirect

Science of Computer Programming

journal homepage: www.elsevier.com/locate/scico

Quantifying IT estimation risks

G.P. Kulk*, R.J. Peters, C. Verhoef

VU University Amsterdam, Department of Computer Science, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 16 March 2009
Received in revised form 27 August 2009
Accepted 1 September 2009
Available online 10 September 2009

Keywords:
Quantifying IT risks
IT-enabled business investment
Logistic regression
Cost overrun
Cost underrun
Risk drivers
Risk factors
Lift chart
Forecasting errors
Data plausibility
Data quality
Cost estimation
Misestimation
EQF
Maturity mismatch

ABSTRACT

A statistical method is proposed for quantifying the impact of factors that influence the quality of the estimation of costs for IT-enabled business projects. We call these factors risk drivers as they influence the risk of the misestimation of project costs. The method can effortlessly be transposed for usage on other important IT key performance indicators (KPIs), such as schedule misestimation or functionality underdelivery. We used logistic regression as a modeling technique to estimate the quantitative impact of risk factors. We did so because logistic regression has been applied successfully in fields including medical science, e.g. in perinatal epidemiology, to answer questions that show a striking resemblance to the questions regarding project risk management. In our study we used data from a large organization in the financial services industry to assess the applicability of logistic modeling in quantifying IT risks. With this real-world example we illustrated how to scrutinize the quality and plausibility of the available data. We explained how to deal with factors that cannot be influenced, also called risk factors, by project management before or in the early stage of a project, but can have an influence on the outcome of the estimation process. We demonstrated how to select the risk drivers using logistic regression. Our research has shown that it is possible to properly quantify these risks, even with the help of crude data. We discussed the interpretation of the models found and showed that the findings are helpful in decision making on measures to be taken to identify potential misestimates and thus mitigate IT risks for individual projects. We proposed increasing the auditing process efficiency by using the found cost misestimation models to classify all projects as either risky projects or non-risky projects. We discovered through our analyses that projects must not be overstaffed and the ratio of external developers must be kept small to obtain better cost estimates. Our research showed that business units that report on financial information tend to be risk mitigating, because they have more cost underruns in comparison with business units without reporting; the latter have more cost overruns. We also discovered a maturity mismatch: an increase from CMM level 1 to 2 did not influence the disparity between a cost estimate and its actual if the maturity of the business is not also increased.

1. Introduction

It is common knowledge that investing in IT is often a risky undertaking. Frequently the targets set at the start of a project are not met, budgets are stretched, IT key performance indicators (KPIs) are grossly underestimated and business KPIs overestimated. IT has an amazingly poor reputation for estimating the costs, effort and duration of IT projects. We can think of cost and schedule overrun and not delivering the requirements agreed upon. A considerable number of IT investments even fail completely. Investing in general depends on, for example, the expected return. The more precisely you can project

* Corresponding author. Tel.: +31 205987782.
E-mail addresses: erald@few.vu.nl (G.P. Kulk), rjpeters@cs.vu.nl (R.J. Peters), x@cs.vu.nl (C. Verhoef).

0167-6423/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.scico.2009.09.001

the probability distribution of the return on investment (ROI), the better you can predict whether an investment makes sense at all. The probability distribution of the expected return is a representation of the returns of individual projects that are part of a portfolio. Misestimating the Bermuda Triangle of project management [13], namely incorrectly estimating the costs, the schedule and the functionality to be delivered, results in a false picture for the expected return, and therefore a false picture of its probability distribution.

Incorrect estimates lead to unexpected results and therefore incorrect estimates of ROIs. Attempts to get an objective assessment of the problem of misestimation have languished while researchers pursue other areas of project measurement. It is not a coincidence that the risk of cost misestimation is an *IT project risk* or for short IT risk. We will propose how to quantify IT risks in such a way that proper IT investment management can emerge.

We deal with the risk management of IT-enabled business investment projects. We speak of an IT-enabled business investment project if at least 25% of the project investment is spent on IT activities. Our research shows that you can analyze investments in IT-enabled projects through the assessment of the IT risks involved. But what are typical IT risks, and how can you quantify them?

We studied other fields where similar questions were treated and solved. A field with a striking resemblance to the IT world is that of perinatal epidemiology [10,9,27,12]. Perinatal means around the birth; epidemiology refers, e.g., to mortality rates for diseases. Perinatal epidemiology deals with, among other issues, mortality rates around the birth. In perinatal epidemiology one searches for easily detectable indicators that will predict with high accuracy the mortality chances for just born children. Historical information is used in such predictions. It will not be a surprise that prematurely born children have a higher mortality rate than other children. There are also differences between boys and girls in this respect. Suppose the medical records of two just born children are shown to you, and you have to predict the mortality chance. One seems to have a few small problems, and the other is prematurely born. Without more precise quantifications, the first has a higher chance of surviving. Let's make this problem more complex. A boy, born after 26 weeks, and a girl, born after 25 weeks, are brought into the hospital. They are both prematurely born, and girls are stronger than boys, but the boy is already a week older than the girl. Now what? It is no longer possible to make a prediction without a precise model that quantifies mortality rates based on indicators like gender and the gestational age in weeks, i.e. the number of weeks dating from the first day of the mother's last menstrual period.

For IT projects similar, but less life-critical, situations exist: most IT projects suffer from misallocated budgets due to the misestimation of the project costs, also called plan inaccuracy [20]. But how does one rank projects by increasing risk, given their status? This is not really possible without decent quantifications. We will show how methods used in perinatal epidemiology transpose to the world of IT to answer such questions. With the quantification of risk, it becomes possible to quantify the expected return of a portfolio. With an ordering of projects by increasing risk, IT investments in the need of management attention will surface and audit attention is optimally allocated. More general, we state that by quantifying the IT risks for an entire IT portfolio it becomes possible to quantify the aggregate expected return of the IT portfolio, and thus it becomes known whether investing in the IT portfolio makes sense. Moreover, it becomes possible to identify the risk drivers and manage new projects based on the right values of the risk drivers with a positive influence on the correct estimation of IT KPIs. We will make the first steps towards achieving these goals by showing how to quantify and find the risk drivers of cost misestimation. Knowing the risk driver misestimation, better estimates are possible in the future, thus supporting investment decisions.

We will focus on quantifying IT risk solely. For quantifying IT value we refer to [49] and to [39], for dealing with IT risks and outsourcing deals we refer to [50], and for aggregating IT investment management to the IT portfolio level we refer to [47]. In an IT governance context IT risks are also of predominant importance. For results concerning IT risks in the light of quantifying IT governance effects we refer to [51], and for connecting IT risk with quantifying IT productivity we refer to [48]. A deep analysis on the risks of requirements volatility is published in [32].

We will deal with risks based on real-life data within a financial services organization. In particular we will propose a method for coming to grips with a very important risk indicator for IT projects: misestimation of costs. An outcome within a selected boundary around the estimated costs is considered a correct estimate. An actual situated outside this boundary is considered a cost misestimation. This method can be effortlessly used to manage schedule misestimation or solutions underdelivery, if for the latter project size measures, e.g. function point countings, are available. The reason for considering these IT risks is that the problems surrounding those indicators are already astonishing, and getting a handle on those risks alone drastically improves the current situation of endemic capital destruction, for many problems with IT projects are caused by the misestimation of IT KPIs. For example: on an annual basis we invest about 290 billion dollars in *aborted* IT projects: about 150 billion in the USA and 140 billion in Europe [18]. For comparison, the annual cost of collapsing buildings in the USA is about 4.4 billion dollars. Collapsing IT projects cost a factor of 34 more.

The reason we did not solely consider overruns, but misestimation in general, is because extensive cost underrun will result in unused money, but this money is put in a reserve. Such unused resources could otherwise have been invested profitably, e.g. in other projects.

The cost consequences of misestimations are substantial and change the picture of the business case of the investment proposal dramatically as shown in an earlier paper [39]. The impact of time overrun and the overrun of estimated costs on the outcome of the Net Present Value (NPV) calculation are quite substantial, depending on the business case.

We stress that project failure, costs, schedule and functionality misestimation risks are not just a case of force majeure, but these risks can be influenced if the risk drivers are known by project management. By taking appropriate measures

to improve the estimation process, the organization can decrease cost misestimation, duration misestimation, and poor solution delivery and increase quality. However, whether this makes sense clearly depends on the time and costs of taking such measures and the impact that these measures will have. We propose the initial steps by explaining how to diagnose the presumable causes of misestimating important IT KPIs.

In this paper we consider the quality of the estimation process and search for the drivers of the risk of the misestimation of project costs. We will analyze the project data using the logistic modeling technique. Logistic regression is a modeling technique which has been applied successfully, as stated, in medical science to answer similar questions. But logistic regression is also a common statistical tool in other sciences; for instance it is often used in marketing to investigate consumer choice behavior. In marketing, the binary variable that is researched represents the buying or not of a certain product [3,5].

In other computer science disciplines in which classification problems arise, logistic regression is also used—for example, in neural networks that are used in bankruptcy prediction, or fraud detection of credit evaluation; for an overview see [36]. Other areas mentioned in [36] for which logistic regression is used in neural networks are engineering, manufacturing and marketing. In the area of software fault prediction, logistic regression is also used (see papers [44,19]), to predict errors before testing. Article [31] uses logistic regression to identify explanatory variables of fault prone software modules over subsequent releases.

Logistic regression allows us to quantify the effect of a particular risk factor on the likelihood that a particular unfavorable outcome, in this paper's context misestimation of IT costs, will occur. Comparable research is conducted in [52], in which schedule estimation is analyzed with logistic regression. But in that paper, the risk drivers are based on surveys; the data that we use in this paper as potential risk drivers are not perception data. In paper [52], logistic regression is not elaborately explained, but the risk drivers resulting from their research are that project managers need to be involved in schedule negotiations and adequate requirements information needs to be available at the time of estimation, and staff need not to be added late to meet aggressive schedules. The latter two both imply a not unreasonable amount of requirements creep during development. A method for quantifying requirements creep effects is presented in [32], illustrated with examples from different industries. In [56] neural networks and logistic regression are compared as early warning systems for predicting project escalation, but it also depends on perception data from surveys. Besides understanding the requirements, also planning, monitoring and controlling the project resulted as significant variables in [56]. Another paper [40] also uses perception data to identify software development success factors by using logistic regression, but also uses a rather small data set of 40 projects. Our research does not use perception data, but real-world project information, such as the estimated budgets and the actual costs.

The reason for considering the particular risk of cost misestimation is not merely a random choice, but inspired by many daily news headlines concerning IT projects. Often huge budget overruns are mentioned, demanding explanations from the responsible manager or government. Almost as often the root causes of these overruns cannot be pinpointed.

Therefore, risk models addressing these risks are useful and needed. Given our quantifications, organizations can focus on the drivers of risk misestimation and move into a position to trade-off whether or not costs are reasonable for mitigating or eliminating the influence of risk drivers and thus increasing the expected yield [39].

1.1. Organization of the paper

The rest of this paper is organized as follows.

Risk drivers and their levels. We start in Section 2 with a discussion of the risk drivers that influence the likelihood of misestimation of project costs. From the literature a large number of IT risk drivers are known. However, which risk drivers should be considered in a specific case heavily depends on the data available within an organization. We describe the process of selecting the risk drivers. Not all project variables need to be considered as risk drivers—for instance, if causality between the binary outcome variable and explanatory variable is lacking. In other organizations different data may have been captured over time, leading to the selection of other risk drivers, and also in that case our approach of selecting the risk drivers is applicable. We also discuss the measure level of the risk drivers. With respect to their measure level the risk drivers range from nominal, categorical, ordinal and interval variables to ratios.

Real-life data. Readers who are interested in the structure of the data set of our real-life case study are directed to Section 3. Of course, all the data have been made anonymous for confidentiality. We show in detail how in general the plausibility and quality of the available data are assessed. The analysis on the reliability and completeness of the data is an essential step in the modeling process, which should always be carried out. Subsequently, we show in Section 3 how we assessed the homogeneity of the data and we pay attention to the test on dependencies between variables. We demonstrate how to select a subset that combines different business units and project types, but still contains enough data from all different types. This set will be used in our subsequent analysis.

Logistic regression explained. In Section 4 we explain the basic ideas behind logistic regression to IT portfolio managers, who do not have any particular statistical and mathematical background. We discuss the striking resemblance of answering questions in the field of clinical research to IT investment decision making. Both phenomena are expressed categorically as dichotomous situations, e.g. yes or no; dead or alive; misestimation or no misestimation. After that, we dive into a more thorough mathematical treatise on logistic regression and show how the parameters of the logistic regression equation are estimated by applying the maximum likelihood principle.

Risk model building. In Section 5 we apply logistic regression to model IT misestimation risks. This results in models based on different misestimation intervals. We discuss the stability of the models as well as the interpretation of the risk drivers found in the models. We discover that the ratio of external developers needs to be kept small and overstaffing must be avoided to improve the quality of cost estimates. Moreover, the statistical analysis displays a maturity mismatch. In IT departments with CMM levels higher than 1, the expected improvement in cost estimation is not realized because the business department remains at a lower maturity level.

Validity of the model found. In Section 6 we compare the goodness of fit of the models found for the risk of misestimation and we discuss the predictive value of the risk models that show the highest goodness of fit test scores. We also show by randomly partitioning the data set in a simulation what the distribution of the lift factor of the misestimation model is and compare it with the lift factor of the model we have found.

Digging into the data. Section 7 shows the results if we analyze the causes of the risk of overrun and the risk of underrun – combined into the risk of misestimation – separately. In this section a new risk driver emerges which has opposite effects on underrun and overrun and therefore did not show up previously.

Conclusions. Finally, in Section 8, we conclude the paper.

2. Misestimation risk and its risk drivers

In this section we define the risk of cost misestimation and the concept of quantifiable IT risk drivers.

2.1. Perception of risk

Many IT risk studies concentrate around the perception of risk factors. In the paper [43] three Delphi surveys were deployed to identify a ranked list of project risk factors. These surveys were conducted in three different countries: Hong Kong, Finland, and the United States. The three panels, one for each country, consisted of experienced project managers. The Delphi survey process in the aforementioned study consisted of three phases. In the first phase of brainstorming each panelist made a list of potential risk factors, with at least six factors. The collected risk factors were compared with each other and combined into an extensive list of risk factors (without exact doubles and similar risk factors). In the next step, each panel, independently of the others, narrowed this list down to a more manageable list. Each panelist chose the most important factors, at least ten, from a random set of risk factors based on the previous extensive list. Each factor that was picked by at least 50% of the participants was taken into account in the final phase. The initial list of more than 150 items was now reduced into three smaller lists: Hong Kong ended up with 15 items, Finland with 23, and the USA compiled 17 risk factors. In the last stage each panelist ranked the remaining risk factors for their panel in order of importance, which resulted in an overall project risk-ranking list per separate panel. In order to get an international ranking, a composite ranking was made. All eleven risk factors that were present in all three ranked lists were ordered by their average relative ranks.

While this gives a good idea of what risks factors are perceived by experienced project managers, it rules out risk factors that they do not perceive. One risk factor that was left out of the composite ranking was the lack of effective project management skills. We expect that this is a typical consequence of perception research: they asked experienced project managers. In two countries, this factor was mentioned: Finland gave it the highest rank 1, and the USA ranked it 5. Due to the consensus step in the last phase of establishing the composite ranking, this important factor dropped out. With the methods that we demonstrate in this paper all risk factors for which we have data are considered to investigate whether they are significant or not. So, we prevent that important factors are missing.

A number of success and risk factors are provided by various authors and consulting companies—for instance, the top 10 success factors mentioned by Standish Group [22–24] or Capers Jones' twelve characteristics of successful IT projects [29]. We summarized these risk drivers elsewhere [50]. Other factors often mentioned are whether new technologies are being used in a project, whether the project managers are experienced and for instance how many management layers are present in a department. What these success and risk factors share is that some of them are not easy to measure, especially not in the dawn of an IT project. And some of them adhere to the perception of risk, rather than the actual risk potential. Let us give a few examples. Jones formulates a success factor *effective communications*. While we all agree that this will surely help in turning any project into a success, it is difficult, but not impossible, to quantify it. When are communications effective? You will only know when a project failed due to ineffective communications. Moreover, such project data, when obtained, are subjective, since such data are usually collected through surveys with project managers. But it is plausible that even subjective data consisting of roughly quantified project information can lead to interesting research results, and when available, should be considered for inclusion in risk analyses. Proper care must be exercised if bold conclusions are drawn that are induced by subjective factors.

Standish Group formulates as the most important project success factor: *executive support*. Again, without doubt, it helps if important people make things happen that otherwise take much more effort. But it is hard to quantify this support, and its effectiveness is hardly predictable. In one company, a student carried out an analysis to figure out whether success and risk factors known in the literature did correlate with actual success and failure. It turned out that none of the factors correlated with the actual situation. For obvious reasons, the data for this case study are confidential, but the result gave us another boost to dive deeper into the problems of how to design reliable models that do predict IT risks. One of the problems of the thesis was that it was hard to quantify some of the factors in an *objective* manner, leading to arbitrary results.

An obstacle with most risk factors that are mentioned in the literature is that they refer to software development projects. The projects under study in this article are not pure IT projects, but IT-enabled business projects. In an IT-enabled business project, the IT part measured in terms of cost is at least 25%, so the IT component does play an important role. Therefore, most of the IT risk factors are useful, but not all.

2.2. Problem definition

This paper proposes a method for finding the factors that determine the disparity between a cost estimation, the financial investment in a project, and its actual. All projects that are considered in this research have project costs that are determined by an *estimate* at the start of a project and register actual costs at the end of a project. Before we continue, we present some important definitions concerning misestimations.

Definition 1. A cost estimation e is defined as being *misestimated* if the actual project costs belonging to that estimate fall *outside* the interval $[e-y\%, e+x\%]$, where y and x are real numbers > 0 and $y < 100$. The percentages y and x are determined or chosen by the decision maker.

Definition 2. An estimation method is *any* method that yields an estimate of the actual. These estimates are anything from amounts that are just guessed, to amounts based on the gut feeling of experienced managers or to amounts forecast from sophisticated models.

Definition 3. An estimation method is defined to be *good*, or *acceptable*, at level α if the percentage of misestimates is lower than $\alpha\%$, where α is some number greater than zero and less than 100. A common value of α is 5.

The above definitions do not require nor imply *any* knowledge of the estimation method. All that matters is the discrepancy between the estimate, induced by the estimation method, and the actual.

The percentages x and y will usually be chosen in such a way that the interval $[e - y\%, e + x\%]$ is not too large. When $y > x$, then the attitude of the decision maker is characterized as *risk avoiding*. If $x > y$, the attitude of the decision maker is characterized as *risk seeking*. In some cases x and y will be chosen equal. In that latter case the loss of underestimating, and reserving money that could have been used in other projects, is judged to be equal to the cost of overestimating and thus creating the need to reserve additional money.

The simplest estimator of the probability of misestimation is obtained by counting how many projects in the historical database displayed cost misestimation and dividing the result by the total number of projects.

2.3. IT risk factors

A very simple example of an exogenous factor in ice cream selling is the weather. It is not possible to change the weather to increase sales, but most probably sales stay lower during overcast and rainy weather than on a bright and sunny day. An example of an exogenous factor which potentially influences the risk of making a misestimation of the costs is the project category to which an investment proposal belongs. In our case study there are four categories: transactional projects, informational projects, strategic projects and infrastructure projects. Clearly, the decision maker cannot change the type of an investment project. Nevertheless, the non-influential factors, like the weather for the ice cream seller, need to be taken into account in decision making. It is important to know for IT decision makers whether or not the risk of misestimating the costs is higher for a certain investment category and how much higher that risk is.

If the quality of the estimation method is improved by changing the value of a risk driver, the chance or probability of making a misestimate will decrease as a consequence. Needless to say, the highest potentially attainable goal is to reduce the chance of misestimation to zero. But this will never be accomplished as some amount of uncertainty will always remain. However, if the quality of the estimation method improves, an excellent job has been done. Although decision makers cannot change the value of an exogenous factor it is very important for them to know the influence of such a factor.

The factors that influence the risk of making misestimates are called risk factors. These factors are called risk factors as they influence, negatively or positively, the risk of making a misestimation. We make a strict distinction between risk factors of which the values can be changed by the decision maker, the risk drivers, and the risk factors of which the values cannot be changed. The factors of the first category will be referred to as *endogenous factors*, but they can also be found called *controllable factors*, *influential factors*, or *risk drivers*. We will refer to the factors of the latter category, which cannot be changed, as *exogenous factors*. These can also be named *uncontrollable factors*, *non-influential factors* or *extraneous factors*.

There is an abundance of literature that provides us with checklists for addressing and recognizing risks in IT. First and foremost there is the qualitative approach of McFarlan [34], who published in the 1980s his first version of a still used extensive questionnaire for addressing risks in information systems. The problem with such questionnaires is that they measure IT risk perception, not the actual risk itself. Of course, they do help to address actual risks since a checklist forces you to think of aspects that would otherwise go unnoticed until the risk materializes.

2.4. Quantifiable risk drivers

Our focus in this paper is on potential quantifiable risk drivers for which we have data available. The following factors are examples of influential and quantifiable risk drivers of IT-enabled business investments.

- Size of the pure IT part of the project, measured in some objective manner, like using function points [1,2,21], lines of code, number of applications, etc.
- Staffing essentials, like the number of FTEs working on the project, ideally broken up into activities.
- Project constraints, like a fixed date or a fixed price contract, internal, external, etc.
- Is a formal development or project management methodology being used? Notice that this is a dichotomous yes/no variable.
- The capability maturity level of the software development department or organization, which is generally known as the CMM or CMMI level.
- Scope aspects of the project, e.g. the number of milestones in the project or the number of changes in the user requirements, or the volatility of the scope expressed as a percentage of change per month [32].
- The experience level of the project team, measured by the total years of experience or by their historical percentage of successful projects.
- The experience level of the project manager, measured by the percentage of successful projects or total years of experience. We can categorize this information in two or three time slots. Think of zero to five years of experience, five to ten, or more. Again, this is a discrete variable.
- The experience level of the user, e.g. the percentage of successful projects completed for a certain user.

Of course, the aforementioned risk drivers are hunches and we do not know whether they are truly influencing the risk that we wish to get a grip on: the risk of IT cost misestimation. And of course, the resulting risk drivers depend on the quality of the data available within an organization. For instance, if there is no information on the experience levels of team members, we cannot examine this risk factor for whether or not it has any influence. For instance, the shoe size of all IT employees is an easily retrievable measure—but, as common sense suggests, not a potential explanatory variable.

In this study we investigate which potential risk drivers, for which data are available, turn out to be real risk factors, given a certain organization and its data. Subsequently, for new projects, it is possible to collect quantifiable data, and feed the data into the established models, and a quantitative impression of the risks is obtained. If the resulting calculated risks are unacceptably high, measures to mitigate the risks need to be taken. Of course, when you address risks, money is involved, and ruling out every risk will lead to exploding project costs. So given the residual risks that are left in the project, it becomes possible to carry out a risk-adjusted appraisal of the value creation of the project. In this paper we will not carry out such risk-adjusted calculations. For elaborate appraisal examples for IT investments we refer the reader to [39,49].

3. The available data

In many organizations there are almost no data available for analyzing any aspect regarding information technology. For those organizations, our paper serves as an example of what is accomplishable if only data were routinely collected. At the same time, such organizations are also in dire need of tools and techniques for coming to grips with their IT function. In such cases we advise basing the quantitative dimension for IT decision making on models that take industrial averages into account. Those models – also for certain important IT risks – are treated in papers directed towards organizations without proper data collection [32,39,47,46,50,49].

In this paper we treat the situation in which data are available. In such a situation there is usually a structured process of data collection, analysis, and feedback. This process is fully integrated within the organization and often there is a special corporate IT department in which data are gathered, combined, scrutinized, analyzed and translated into strategic information for executive decision makers. The organization from which we obtained data is divided into reporting units from which individual and aggregate information is collected.

But also in this case there are limitations and constraints. In our study we used data from finished projects. Finished projects provide, next to their estimates, information on actual performance with respect to actual costs, actual project duration and delivered functionality. Using such historical data, plus a number of potential explanatory variables, we analyzed which variables correlated with which IT risks, if at all.

Next to project specific data we also have access to data describing the context in which the projects were carried out. In Table 1, we give an overview of the most prominent generic information that is often easily available at the reporting unit level. This information is also available in our data set that we will analyze later on.

Let us explain Table 1. A reporting unit is some logical part of an organization, either in terms of business, geographical dispersion, cross-cutting concern (e.g., security) or otherwise. An executive center is a collection of such reporting units, e.g., the Ministry of Homeland Security. Typical names for such collections are well-known. EMEA, an acronym for Europe, Middle-East and Africa, is a typical reporting unit that is found in many organizations. Regional or logical collections of reporting units are not necessarily organized accordingly.

A line of business is also a well-known term, easy to identify, and characteristic for a reporting unit. Think of MRI scanners, mobile phones, private banking, pension administration, etc. as lines of business within organizations. Reporting units spend money, and contain staff members. This organization routinely collects a number of such aggregates. Total IT costs is an aggregate that comprises the entire annual IT activities within a reporting unit. Think of the IT budget of the FBI, or the IT budget of the Department of Housing and Urban Development. Total IT staff is the pendant of total IT costs: it aggregates all staff members of a reporting unit that carry out IT related activities. Furthermore, some breakdowns for staff are shown:

Table 1
Generic variables giving information on the environment in which projects are carried out.

Generic information	Meaning
Reporting unit (RU)	Logical business or regional part
Executive center	Logical collection of RUs
Line of business	Type of business for a RU
Total IT costs	Total IT related costs per RU
Total IT staff (TIS)	Total IT staff per RU
Internal IT staff	Breakdown of TIS
External IT staff	Breakdown of TIS
Management methodology	Is a management methodology used in a RU?
IT maturity level	Capability maturity model level of a RU

Table 2
Project specific variables giving information on project performance within the responsible reporting unit.

Specific information	Abbreviation	Meaning
Reporting unit	*ru*	Business owner of the project, to link generic information to the project
Project category	*pc*	Type of investment project
Estimated costs	*ec*	Estimated costs of the project
Actual costs	*ac*	Actual costs of the project
Estimated duration	*ed*	Expected duration of the project
Actual duration	*ad*	Actual duration of the project
Estimated project power	*epp*	Size of the project in terms of average investment per month
Actual project power	*app*	Size of the project in terms of average investment per month
In-house/outsourced	*io*	Either done largely in-house or outsourced
Functionality delivered	*fd*	Percentage of delivered functionality

internal versus external, to mention an obvious one. Finally two other kinds of variables reflecting quality of IT management and IT craftmanship are listed: whether or not a project management tool is used and the CMM level of the reporting unit. If no assessment has been done on the maturity of the IT process, or it is not more than ad hoc, the CMM defaults to level 1.

Apart from these generic characteristics at the reporting unit level, there is also individual project information. In Table 2 we summarize the most important project specific variables.

We elaborate on Table 2. We already encountered the reporting unit (RU) in Table 1, but in this context it is the business owner under whose authority the project is being carried out and it is denoted as *ru*. It is used to link the generic reporting unit information to a project. The project category *pc* refers to the type of project investment. We distinguish four categories: transactional, informational, strategic, and infrastructure projects. It is often easily determined whether a project fits within one of the above categories, a list that we adopted from [53]. We summarize these four types below.

- Transactional investments: Transactional investments provide the technology to process the basic, repetitive transactions of the business, e.g., transaction processing, accounting, account management etc. The main purpose of this type of IT investment is to improve efficiency and to reduce costs.
- Informational investments: Informational technology provides the technology for managing and controlling the organization. Systems in this category typically include systems for management and financial control, decision making, planning, communication and accounting.
- Strategic investments: Strategic investments will usually be designed to add real value to the business by increasing competitive advantage, enabling the entry into new markets, or otherwise increasing or enhancing revenue streams. Examples are a system for supporting an Internet-enabled business initiative, cable TV-enabled marketing channels, etc.
- Infrastructure investments: Investments in infrastructure are often of long duration and costly, but may not, in themselves, generate any directly quantifiable financial benefits, although the business applications that depend upon the infrastructure can do so. Therefore, the investments in and maintenance of infrastructure are essential, and not always immediately profitable. This is an essential notion, already mentioned by Adam Smith in 1776 [45, book V, chapter 1, article 1]. Examples of infrastructure investments are the implementation of a new or upgraded systems management product (e.g., Unicenter or Tivoli), the implementation of a new operating system (e.g., Linux), or the roll-out of a new, private, network. Most such investments are likely to be non-discretionary.

So, each project is categorized into one of these four categories. More mundane variables are the variable *ec* which is the expected cost, in millions of dollars, of the financial investment for the project until completion and delivery. The actual financial investment in a project is denoted by *ac*, also measured in millions of dollars to ease comparison with the estimated costs. The next variable in Table 2 is the expected duration *ed*, in calendar months, of the project until completion deployment. Such variables in combination with the actual values provide a rich potential for developing predictive models for schedule misestimation. The actual duration, in calendar months, of a project until completion and delivery, is denoted by *ad*. The estimated and actual project powers, which are the sizes of the project in terms of average investment per month,

Table 3
Project specific risk.

Project risk outcome	Abbr.	Definition	Measure level
Cost estimation quality	*cm*	Misestimated if the actual is outside $[ec - y\%, ec + x\%]$, good otherwise	Nominal (0 or 1)

are denoted by respectively *epp* and *app*. Furthermore, it is fairly easy to determine whether a project is done largely in-house or is outsourced; this information is summarized in the variable *io*. The *fd* variable represents the percentage of required functionality that is delivered within a project.

3.1. The research data

The aforementioned variables are classified into response variables and explanatory variables. Each response variable is a project risk, and the explanatory variables are ideally influential, i.e. the risk drivers. As Tables 1 and 2 indicated already, the risk drivers are separated into generic aggregates at the reporting unit level and characteristics at the individual project level.

In Table 3 we define the outcome of the project risk of cost misestimation. In the following sections we will analyze the risk of misestimation of project costs. The analysis of the other risks such as duration misestimation is analogous to the method we describe in this paper.

The quality of the estimated costs, *cm*, is a crude dichotomous metric. In the case where the estimation of the costs was significantly misestimated, then $cm = 1$. The variable *cm* is set to 0 if this was not the case. So, $cm = 1$ indicates that costs were misestimated. On the other hand, $cm = 0$ means that the estimation was within the predefined bounds.

We denote the risk of cost misestimation, which can have any value in the range $[0, 1]$, as the variable p_{cm}. As stated in Table 3, we define a project having a good estimate if the actual costs have a value in the interval $[e - y\%, e + x\%]$ for a certain x and y. For example, let $x = 2.5$ and $y = 5$ and let us assume that 70% of the projects were misestimated. We now want to know which factors were influential on the performance of the estimation process. Which factors can in the future positively influence the outcome of a cost estimate? Does the accuracy of the estimate depend on having a mature organization? Or on using certain project management tooling? The research presented here studies the quality of estimates after the estimates have been produced by some estimating technique. It does not consider the estimation method itself.

We are using an interval to indicate misestimation and since we do encounter underruns and overruns for our chosen interval, the method that will be used to find causes of misestimation, i.e. the method of logistic regression, will be able to find useful results. As in perinatal epidemiology, we will not research the amount of survived days, or in our case the amount of overrun or underrun, but the research interest lies in the survival itself or in our case misestimation. Because we use a bandwidth around the estimate, the model treats actuals close or equal to an estimate in the same way.

We will focus on the risk of cost misestimation in this paper. Our goal is to explain how to calculate such risks. Of course the method is applicable to the other two aforementioned risks as well.

The measure level in Table 3 refers to the kinds of scales and levels of measurement for each variable. As an intermezzo we explain this for the uninitiated. We already alluded to the distinction between discrete and continuous variables. Discrete, or categorical, variables are variables in which there are no intermediate values possible. Continuous variables can theoretically take any value in between two points. The estimation quality is a discrete variable: it is either zero or one. Nothing in between is possible in the world that we defined.

The measure levels for a variable refer to either one of the four basic levels: nominal, ordinal used for categorical variables and interval, or ratio, used for continuous variables. We explain the four levels for completeness.

- A variable measured on a *nominal* scale is a variable that does not really have any evaluative distinction. One value is really not any greater than another. A good example of a nominal variable that we encountered earlier on in this paper is gender: boy or girl. Information about nominal scales is usually coded with numbers. We used the number zero for misestimation and one if this project outcome did not materialize for the project. Of course, this choice is arbitrary and one value is not larger or better than another choice. As illustrated we use these – arbitrary – encodings in all kinds of formulas or programs to do calculations with nominal variables. The main idea is that with nominal variables there is a qualitative difference between values, not a quantitative one.
- A variable that is measured on an *ordinal* scale does have an evaluative connotation. One value is also in reality greater or larger or better than the other. An example is the earlier mentioned perception of risks. You can rate risks on a scale from 1 to 10, with 10 representing no risk, and 1 very high risk. With ordinal scales, we only know that higher is better than lower, only we do not know by how much. Also the scale is not constant, in the sense that the distance between 1 and 3 could vary from the distance between 7 and 10. A well-known ordinal scale is the Likert scale [33], often used as a five-point scale in questionnaires.
- A variable measured on an interval scale gives the same information as ordinal scales do, but interval variables have in addition an equal distance between values. The percentage of cost overrun is a good example: the difference between 10% and 20% cost overrun is the same as the difference between 60% and 70% cost overrun.

Table 4
Classification factors for projects.

Classification factors	Abbr.	Definition	Level
Project category	*pc*	Type of project investment category	Nominal
Line of business	*lob*	Type of business for which the project is carried out	Nominal
Executive center	*ec*	The executive center in which the project is carried out	Nominal

Table 5
Generic risk factors at the reporting unit level.

Reporting unit specific risk factors	Abbr.	Definition	Level
IT maturity	$cmm_{>1}$	RU's CMM level in which the project is done	Ordinal
Reporting unit size	*rs*	Size of RU in terms of total IT costs	Ratio
Project management tool	*pmt*	Project management tool used or not	Nominal
Reporting quality of financial information	*rqf*	Reporting has been done or not	Ordinal
Development department size	*dds*	% of development and enhancement staff in total IT staff	Ratio
Internal development staff size	*ids*	Breakdown of DDS	Ratio

- Variables measured on a ratio scale have the same properties as ones on an interval scale have, but in addition, there is an absolute zero point.

This concludes our little discussion of project measure levels.

For the project outcome *duration misestimation* (*dm*), we have a similar dichotomous distribution: duration misestimation is present if the project experienced more than *x*% overrun or less than *y*% time underrun, and not if this did not occur. For functionality failure we also have a dichotomous definition, but this is a little different than the others. We consider a project to have failed in functionality delivery if less than 95%, the threshold, of the functionality is delivered than was targeted for, or if the requirements creep was so high that more than 110% is delivered. If this is the case we assign 1 to *ff*, and otherwise 0. To ease comparison and calculations we recommend measuring functionality in function points.

Furthermore, we have a continuous variable representing the amount of overrun, or for that matter underrun. Note that the latter is possible if people estimate much too high costs, which happens for instance for political reasons. Estimated project costs can also contain biases, for instance if salami tactics are being used. For an extensive paper on forecasting quality see [20]. This variable is measured on an interval scale and it is the percentage of cost overrun. It is easily calculated by dividing the estimated costs by the difference of the actual costs minus the estimate. The project risk schedule misestimation is analogous to cost misestimation. It considers the overrun or underrun of the project schedule. Functionality misestimation considers either solutions underdelivery, or in the case of overrun, requirements creep if all functionality plus all added requirements have been delivered.

Next, we discuss the non-influential classification factors that we used. Risk factors are called classification factors if they cannot be influenced, as is the case with the weather for the earlier mentioned ice cream seller. We summarized the classification factors in Table 4. We already met the factor *project category*. We encode the four categories with the numbers 1, . . . , 4 as follows: transactional is 1, informational 2, strategic 3 and infrastructure is assigned 4. Since 1 is not better or different than 3 this is a nominal scale. The other two classification factors were already discussed before.

This brings us at the other risk factors: the generic risk factors that are playing a role at the reporting unit level, and the project specific risk factors. We start with the generic ones in Table 5. We used the reporting unit's maturity level: the CMM level, the size of the reporting unit in terms of its total IT budget, and whether or not an overall project management tool was used within a reporting unit. Instead of using the CMM levels verbatim in the analyses, we transformed this variable. If a reporting unit has CMM level 1, the variable $cmm_{>1}$ has the value 0; if the reporting unit has level 2 or 3, the variable $cmm_{>1}$ has the value 1. We do this because there are very few instances of level 3 in our data set. Too few instances of a level can easily lead to uninterpretable or wrong conclusions later on. Furthermore, we used a rating on the quality of the reporting unit on financial information. Often this is known from internal or external audits done by accountants, and it is complemented with other indicators, like reporting to the corporate IT department about the financial state of the IT-enabled business investments themselves. Furthermore, we used the size of a reporting unit in terms of total IT staff dealing with functionality change: new and enhancements. We used a breakdown of that generic risk driver and calculated the percentage of such IT staff who were internal.

The quality of the project management is an important risk driver according to the literature [22–24,30]. In the aforementioned references an experienced project manager was ranked high. Actual data for such risk drivers are usually not available or difficult to measure. An indication based on running project data is available in a direct manner. In our field study, the reporting units needed to report on the expected financial performance of running projects. Think of reporting a Net Present Value, an Internal Rate or Return, Economic Value Added, Risk-adjusted Return on Capital, and other economic measures. See [49,39] for such analyses on IT-intensive investments. The presence of these financial performance indicators implies the existence of a proper business case.

The *rqf* risk driver depicts whether a reporting unit has been reporting on financial indicators or not. We distinguish two categories for the variable *rqf*. The levels of this variable are *no reports* for units who have not reported on their expected

Table 6
Risk factors at the project level.

Project specific risk factors	Abbr.	Definition	Level
Estimated costs	*ec*	Size of the project in terms of total estimated costs	Ratio
Estimated duration	*ed*	Size of the project in terms of total estimated duration	Ratio
Project power	*epp*	Estimated size of the project in terms of average costs per month	Ratio
In-house or outsourced	*io*	Project either developed in-house or outsourced	Nominal
Project category	*pc*	Transactional, infrastructure, strategic or informational	Nominal

financial returns for any of their projects and *reports available* for those with a capability of reporting on risk and return data. To give an indication, on the running project data, we encountered about 27% of the reporting units that we assigned *no reports* to, so the majority did report on at least a few projects. Normally, the largest projects obtain more management attention than the smaller ones. So, even if only a few projects have financial reports, it will most probably add up to a large percentage of the total amount of IT investments. That is why we chose to flip the variable to true as soon as there was some financial reporting.

From Table 2 we selected the variables for analyses that are shown in Table 6; variables in Table 2 that refer to actuals, such as actual costs or delivered functionality, are left out, since they cannot influence the quality of an estimate, as the sold amount of ice creams cannot influence the estimated amount of sold ice creams. In the next section we will assess the quality of our data set.

3.2. Data quality

The project database contained information on 221 finalized projects of a large organization totaling to a financial investment of at least $435 million—for 33 projects cost figures were missing. We narrowed the data set down to 165 projects with estimated costs of $370 million by taking the following criteria into consideration.

- Projects with missing data were not included in the research set.
- Double entries of projects were removed from the data set.

The remaining 165 projects have no missing data regarding potential project risk drivers. In a few cases, we completed missing data regarding potential generic risk drivers by copying these data from other projects in the same reporting unit. For two reporting units the CMM level was missing. An experienced IT auditor estimated their CMM levels. The reasons for leaving projects out of the 165-project data set were one or more missing data fields. Almost always success criteria were missing. Missing data on the duration of a project was ranked second as the reason for removing a project from the data set. Furthermore, we noticed cost data missing, functionality data missing, or mistakes like actual costs of zero dollar. We removed three double projects. These were three relatively small projects, and they accounted for only 0.12% of the original estimated costs of $435 million of the 165 projects. Notice that in a situation when you start up this kind of IT portfolio management practice, you can save millions of dollars just by removing truly redundant projects from the portfolio [6,47]. In this case, no projects were done twice for real, they were just reported on more than once.

3.2.1. Data correctness

The collected data are reasonable, correct and reliable. Namely, in this organization there is a data intake procedure, where unlikely values are detected and currency issues are checked and corrected. We can weed out these errors easily. Methods for detecting such values are based on comparisons with industry benchmarks. Let us explain with an example project that is estimated with $60 million project costs and whose estimated duration is about 6 months. We use a formula from [47] that is as follows:

$$tcd(d) = \frac{rw}{1800} \cdot d^{3.564}. \tag{1}$$

In Formula (1), w is the number of working days in a year, r is the daily burdened compensation rate, and d is the duration of the project in calendar months. So, for a given duration we can calculate with Formula (1) its total cost of development $tcd(d)$ according to industry benchmarks. In the above example, we just take d equal to 6 months, and for the specific reporting unit we use the generic data for daily rates and working days per annum to calculate the total cost of development. As an example, we take 200 working days per year and a daily burdened rate of $1,000. This results in $tcd(6) = 65,930.83$ dollar. What happened in the example project is that the local reporting system works in thousands of dollars, and the project that was estimated to cost $60,000 was keyed in as such. Adding an additional erroneous three zeros to the project in the reporting system led to the unlikely high cost of $60 million. In the same way, currency problems and staff problems are detected, and other extreme outliers. Of course, there are many kinds of projects and for other kinds of projects other benchmark formulas were necessary. For an overview of a number of such benchmark formulas and how to create them, we refer the interested reader to [47], in which about fifty formulas are found based on industry benchmarks.

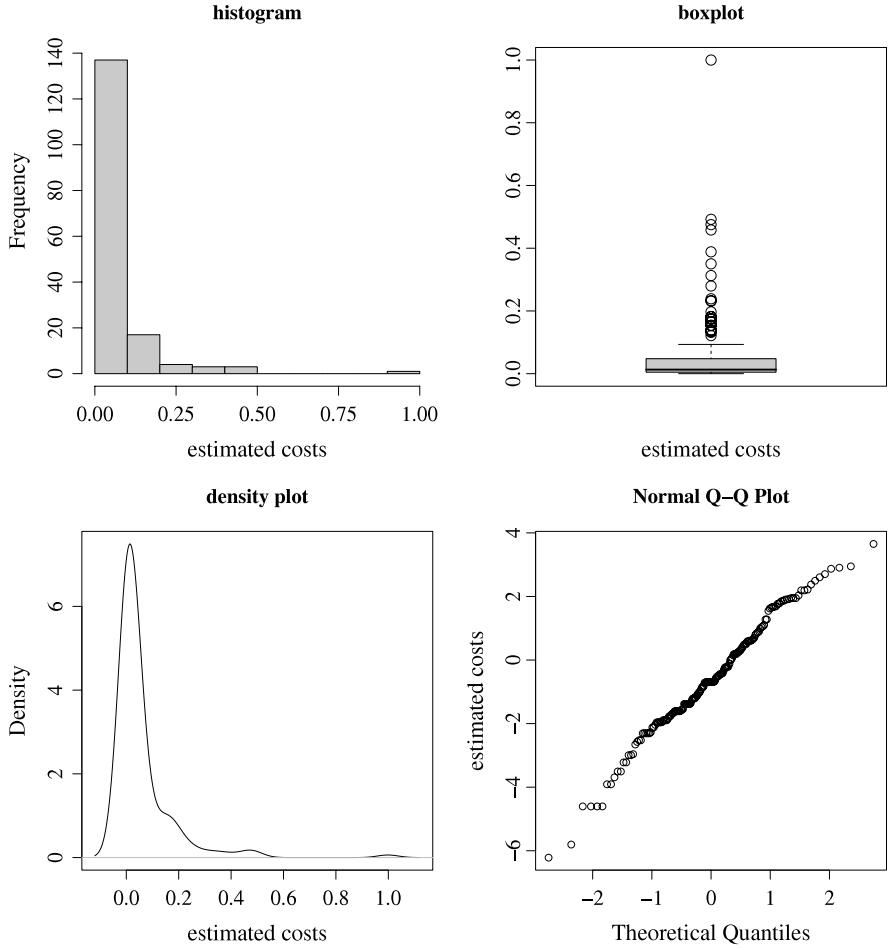

Fig. 1. Visual insight into the empirical distribution of the estimated costs (indexed for confidentiality).

3.2.2. Data plausibility

To have a quick overview of the data we conducted the following visual inspection. In Fig. 1 we display a histogram, a box plot, a density plot, and a Q–Q plot of the estimated costs in the data set.

A histogram displays the frequency of a variable in certain value classes, which gives us a rough indication of the distribution. A box plot represents a graphical sketch of the numerical statistics. The solid box depicts the data between the first and the third quartile, the inter-quartile range, displaying 50% of the data, which is a rather small box in our case. The line within the box represents the median, which cuts the data in half. The so-called whiskers embody the boundaries of the box plot representing the bulk of the data; data points outside these limits are often considered as outliers. A density plot depicts a smooth estimate of the distribution or density. This estimate is based on subparts of the values of variables. Our Q–Q plot, or Quantile–Quantile plot tests the data against the log-normal distribution with unit rate. If this plot is more or less a straight line, then we have an indication that the probability density function belongs to the family of log-normal distributions. In our experience, this is the normal pattern if data have not been manipulated.

Let us discuss the four plots in this figure. First of all we plot a histogram. This one shows that the majority of the estimated financial investments are in the partition with the lowest investments. We immediately see that there are outliers, and they can be pretty large. The next plot is the box-and-whiskers plot. Fifty per cent of the data are cluttered around the median. The horizontal line segments outside the whiskers, the horizontal square brackets, are the potential outliers. There are many potential outliers, all high values. Furthermore, in the lower left plot of Fig. 1 we estimated an empirical probability distribution of the estimated durations in the data set. We notice that this distribution has a spike around the third quartile, and a long tail to the right. The Q–Q plot is roughly a straight line, giving us further visual evidence that the data has a log-normal distribution. The above analysis has been carried out for other variables also: actual costs and estimated and actual duration. These analyses showed the same result, that these KPIs have a log-normal distribution. IT KPIs often display long tails in their distribution and a log-normal distribution has a long tail. So, the data are in line with characteristics that we usually encounter.

Comparing the distributional behavior of the estimated durations with the estimated costs, we expect somehow a correlation. As elaborately treated in [48], these correlations are almost always not representing a one-to-one

correspondence, since a lot of stochastic effects are in place when constructing IT-enabled business projects. An exception to this general rule is the case of the largest outliers. And also in this case, the single outlier above 60 months and the duration of the largest project are from the same project. Another interesting observation is that there are somewhat fewer outliers of the estimated duration, and they are more evenly scattered, than for the estimated investments. This is an indication that the style of IT governance in this organization is more directed towards managing on costs than on time. In an organization that manages uniformly on time, you would expect to see clouds of data around certain time frames, like 12 months. Also these aspects are useful in the analysis of distributional behavior of important KPIs of IT-intensive projects; such patterns are elaborately discussed elsewhere [51].

3.2.3. Overperfect data detection

So far, we have shown that the data in our data set of 165 projects are plausible in the sense that the KPIs display plausible characteristic distributional behavior. But we have to take overperfect data [51] into account. Overperfect data are data that are too good to be true. Suppose you plan for certain KPIs, say durations, or costs. Then it is important to know how good the predictions are. You can do this by comparing estimated and actual KPIs. Sometimes the similarities are so striking that the chances that the estimated data are retrofitted to the actual data are very large. In order to spot such effects we compare the estimated KPIs against their corresponding actuals. We do so by calculating the correlation coefficient of the estimated and actual costs, r^2, which has a value of 0.98. If the correlation was 1, than all actuals would have been equal to the estimates or transformed by a constant value, a case of overperfection.

In order to obtain more confidence about this, we overlaid their cumulative distribution functions, which are the integrals of the probability density functions. We notice that the two lines almost coincide. So, the hunch that the distributions are potentially similar turned out to be valid. Now is this a case of overperfect data, or is this a case of very good planning? From the data we cannot tell the difference. So additional qualitative insight is needed here. In all cases where data show such similarities it is wise to dive into the reasons. In this case it turned out that the estimated data cannot be retrofitted once the data are reported to the corporate IT department and the data are checked for unlikely values. Additionally, since the data collection is quite mature, the data producers gained a lot of experience in estimating the costs correctly. Since the costs in the data set were rounded off to thousands or millions of dollars this also led to actuals being equal to estimates. Although this data set is very suitable for the analysis presented here, more exact, not rounded off, numbers are always preferable for data analysis.

We learned from all these visual statistical plots that the KPIs under investigation are reasonable and plausible. A method for measuring the quality of an estimation method, and analyzing for political influences or other biases, is presented by Eveleens and Verhoef [20]. The method uses the Estimating Quality Factor (EQF) and they present benchmarks for the EQF. The median EQF of the estimated costs for our data set is 9.4. This is a rather good result since in their paper a median EQF of 8.5 is considered good estimation practice. Although the quality of the estimation practice in our case study is not bad at all, there is still room for improvement. By defining a cost misestimation as actual project costs falling outside the interval [−5%, 2.5%], a considerate amount of cost estimates are classified as misestimates. Therefore, it still makes sense to search for the risk drivers of the misestimations.

3.3. Pooling or splitting up?

Now that we are confident that the useful data are also plausible, we address another question: can we pool all projects into one data set to analyze for the influence of risk factors?

As a first check we created Fig. 2 to inspect the distribution of the disparity between actual costs and estimated costs. This figure displays that cost overrun as well as underrun occurred and that there are no multiple peaks in the distribution. Multiple peaks are an indicator that the data set should be split up into two or more sets each containing one peak. Each peak has its own causes and needs to be analyzed separately. Since this is not the case, we continue with the entire data set.

Our data set is a so-called pooled data set: the original data are collected from various different samples; in this case from different reporting units. We must know whether the pooled data set is homogeneous with respect to the phenomenon which we want to explain: the variation in the risk of project cost misestimation. If the data set can be subdivided into subsets that are not comparable in kind of nature with respect to the explained variable, the data set is called heterogeneous. Let us explain. In our case study the data set can be subdivided in a number of ways. For example it can be subdivided along the axis of the executive centers. We then obtain three subsets of 25, 29, and 111 projects respectively. For each subset a specific influence on the risk of project cost misestimation can play a role which is specific to that subset which does not apply to the other subsets. Let us label that factor culture. If this culture effect exists, the three subsets are not homogeneous with respect to the phenomenon that we want to explain.

There are two ways to deal with this problem. First, we can pool the data of the three subsets and include in our model a categorical variable which can take one of the values 1, 2 or 3 and include interaction variables that allow for the combined influence of this categorical variable and other variables. If a specific EC-culture influence exists, the categorical variable and/or some interaction variables will show up in the logistic regression equation with a significant coefficient. Second, we can estimate the logistic regression equation for each of the three subsets separately. In that case we do not have to include a categorical variable and interaction variables in the three models. The result will be three different logistic regression

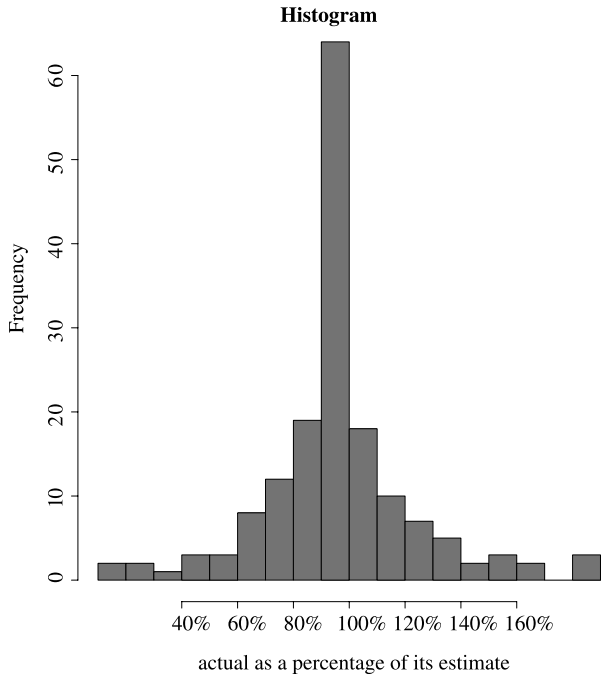

Fig. 2. Histogram of the disparity between actuals and estimates of investments.

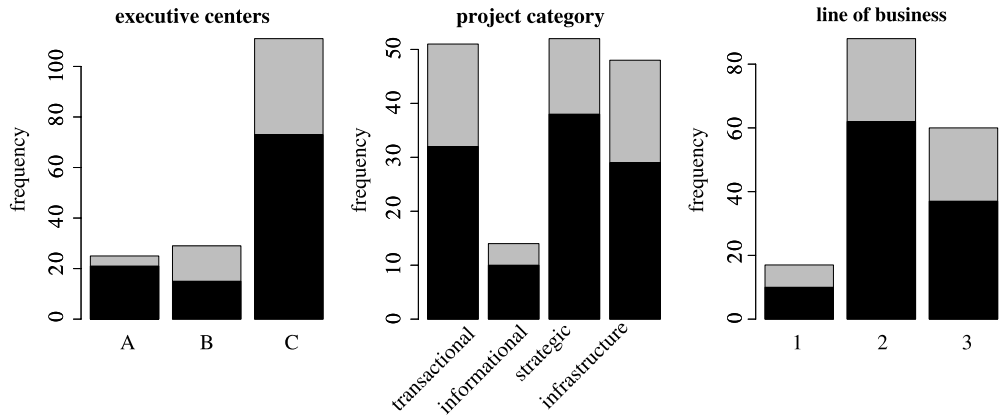

Fig. 3. Distribution of misestimation among potential risk factors.

equations for the three subsets. However, in both approaches the number of observations in each subset must be sufficiently large to carry out meaningful statistical analyses. If different causes for project cost misestimation exist across the executive centers, and if we want to search for these differences, we need enough projects of each subcategory. If a subset contains not enough data points in relation to the number of explanatory variables, it is statistically not possible to test on heterogeneity. Or, in other words if a subset does not contain enough projects it is not possible to determine whether there is a culture effect which must be taken into account in explaining the variation in the risk of misestimating project costs.

Of course, we can also subdivide the data set along the axes of lines of business (LOB), and project category (PC) to investigate the presence of LOB and PC effects. Potential subsets based on these two classification factors are:

- Subsets by line of business: In our case there are three major lines of business, with 88, 60, and 17 projects.
- Subsets by project category: In our data set there exist four categories: 51 transactional projects, 14 informational projects, 52 strategic projects and 48 infrastructure projects.

In Fig. 3 we depict simple bar plots of the three classification factors that are present in our sample set. In the figure we include the amounts of misestimated projects for each category.

We notice that the bar plots display similar distributional shapes for well estimated and misestimated projects. Only slight differences for the categories with small sets of data are displayed, which differences are very likely to be induced by the small size of these data sets. But the estimation process can have differences between executive centers and also in project category or line of business.

Table 7
Three-way table of the exogenous variables and the percentage of misestimation for each class.

Project Category		1		2		3		4	
	estimate	% mis	n	% mis	n	% mis	n	% mis	n
EC	LoB								
A	1	100%	2	-	0	100%	1	0%	2
	3	100%	2	100%	6	89%	9	67%	3
B	2	75%	4	-	0	100%	1	-	0
	3	50%	8	0%	1	57%	7	38%	8
C	1	100%	1	50%	2	80%	5	25%	4
	2	62%	29	75%	4	73%	22	75%	28
	3	40%	5	0%	1	57%	7	67%	3

However, we are interested in the possible combined EC/LOB/PC effects on the risk of project cost misestimation. If we subdivide the data set along the axes of the ECs, LOBs and PCs we obtain $3 \cdot 3 \cdot 4 = 36$ subsets. Each subset differs from the other subsets by the influence of a specific EC/LOB/PC combination, that holds for that subset and does not apply to the other subsets. In Table 7 we present the number of projects and the percentage of misestimation per subset.

It is questionable whether all subsets do contain enough data to estimate the logistic regression equation for that influence of that subset. Table 7 shows that certain lines of business do not occur in certain executive centers and that the three executive centers and the three lines of business show differences in the misestimation percentage. Therefore, it will be hard to determine statistically whether or not the specific EC/LOB/PC combination has an influence on the risk of project cost misestimation. We will not dive into the sample size issue, but restrict our attention to the largest subset which contains enough projects of the different subcategories that are left. This results in the subset defined by executive center C and line of business 2, shaded in Table 7.

We see similar misestimation percentages in all four project categories and these categories contain sufficient projects, except for project category 2. The projects in this executive center and line of business still have a substantial amount of data points: 79 projects. Therefore, we will continue our search for the influential factors of misestimation with the data set of projects in executive center C, line of business 2 and project categories 1, 2 and 4. This is a data set that contains enough data points in each project category class for searching for influences of the potential risk factor under consideration.

Note, that there are more classification variables than *ec*, *lob*, and *pc* among our risk factors. We refer to the variables *io*: project either developed in-house or outsourced, *pmt*: project management tool used or not, *rqf*: reporting on financial information or not, and $cmm_{>1}$: CMM level higher than 1 or not in the project's reporting unit. These variables are dichotomous variables that can just take two values: 1 or 0. Since we have enough projects in our subset of 79 projects of each type, we do not have to split up the data set further. The logistic regression technique includes dummy variables in our logistic regression model that allow for the potential influence of *io*, *pmt*, *rqf*, and $cmm_{>1}$. If some dichotomous variable has an influence the variable and/or some of its interaction variables will show up in the logistic regression equation with a coefficient that differs from zero in a statistically significant way.

As stated, a way to further split up the data is to leave out all the outsourced projects (*io*). Outsourced projects must be left out if the risk of misestimation depends on the characteristics of the company to which the system development is outsourced, as opposed to a dependency on the company that outsources the labor. In the case of dependency on the company to which the development is outsourced, we need information about the CMM level of that company, their project management tools, and so on. For our data set, this is not the case since the estimates were made in-house and not by the outsourcing party. Therefore, estimates are dependent on the maturity level and the project management tools of the company itself. Later, we will reconsider this issue of pooling in-house and outsourced projects. As stated, if a difference in misestimation for in-house and outsourced projects exists, the variable *io* will show up in the logistic regression equation as a risk driver.

3.4. Dependencies between variables

Next, we investigate whether the variables in our remaining data set of 79 projects display mutual dependent behavior. This is important, because when having strongly dependent variables, one of them could show up in the analysis, and the other not. In that case we have to check which of the two is the explanatory variable. We distinguish three kinds of dependencies, because our set consists of continuous and categorical variables.

- Dependencies between pairs of continuous variables.

Table 8
Matrix of correlation between continuous risk drivers.

	ec	ed	epp	rs	dds	ids
ec		0.75	0.72	0.55	0.26	-0.2
ed			0.31	0.49	0.18	-0.29
epp				0.57	0.3	-0.19
rs					0.42	-0.2
dds						-0.07
ids						

Table 9
Contingency table with actual number of IT-enabled business investments grouped on maturity level and whether or not a project management tool is in use.

$cmm_{>1}$	$pmt_{\text{not used}}$	pmt_{used}	Total
1	26	27	53
> 1	6	20	26
Total	32	47	79

- Dependencies between pairs of continuous and categorical variables.
- Dependencies between pairs of categorical variables.

We start to inspect the interdependencies of the continuous risk drivers. To that end we calculated the correlations between continuous variables. A perfect correlation is displayed by the number 1, a perfect negative correlation is displayed by -1, and no correlation by 0. Numbers in between indicate low or high correlation depending on the displayed number. The lower left half of Table 8 is left empty for readability, but can of course be filled with a mirrored copy of the upper right half of the table. We spot immediately that not all risk drivers are independent. We discuss the dependencies. The estimated investment, *ec*, correlates with the estimated duration, *ed*, this is displayed by the value 0.75 in Table 8. This is what you hope to be the case, because projects with long durations usually have higher costs than short projects, where the latter display lower costs. The ratio of development and enhancement staff to the total IT staff, *dds*, does not correlate with the ratio of internal development staff *ids*, as is displayed by the value -0.07. If in the logistic regression analysis two risk drivers emerge that are strongly dependent, we need to consider whether one of the two variables can be left out.

We now consider tests of independence for pairs of categorical variables, for which we use contingency tables. A contingency table represents the combined counts of the levels of two categorical variables. With the well-known statistical χ^2-test we then infer whether two variables are dependent.

We illustrate the use of this χ^2-test with an example. We investigated the potential dependency between the dichotomous variable indicating a higher maturity level than level 1, $cmm_{>1}$, and whether the reporting unit uses a project management tool. Table 9 depicts the observed amount of projects for each combination of the maturity level $cmm_{>1}$ and the variable *pmt* indicating whether a project management tool is in use.

To establish whether the two factors are independent, we calculate the expected amounts of projects in each category based on Table 9. There are in total 79 projects of which 53 projects reside in a reporting unit with CMM level 1, so the chance of having CMM level 1 is $53/79 = 0.67$. Since there are in total 32 projects that are in a reporting unit without a project management tool in use, the frequency of $pmt_{\text{not used}}$ projects with CMM level 1 should be $32 \cdot 0.67 = 21.5$. In this manner, the expected cell counts are estimated as the products of the observed marginal totals divided by the table total. In this way we can fill another contingency table on the basis of the assumption that both variables are independent. Table 10 contains all the expected amounts.

The Cochran conditions [15,16] are used as a rule of thumb for whether the χ^2-test can be used to test the dependency of two variables. These conditions state that 80% of the expected values in tables calculated like Table 10 need to be higher than 5 and all values need to be higher than 0. For Table 10 we use these Cochran conditions to check whether a χ^2-test should be used at all: none of the cells has an expected amount of zero, and more than 80% of the cells have a value higher than 5. Therefore, it makes sense to carry out a χ^2-test. The test gives a χ^2-statistic of 3.9 and a p-value of 0.049. This calculated p-value is small, so at a 5% confidence interval we can reject the hypothesis that the maturity level is independent of whether a project resides in a reporting unit in which a project management tool is in use. We thus have to assume that the two variables are dependent on each other, a conclusion that makes sense, since, whenever the maturity level of a reporting unit increases, the usage of a project management tool will be required.

Table 10
Contingency table with the expected number of IT-enabled business investments grouped by maturity level and whether or not the reporting unit that creates the estimates uses a project management tool.

$cmm_{>1}$	$pmt_{\text{not used}}$	pmt_{used}	Total
1	21.5	31.5	53
> 1	10.5	15.5	26
Total	32	47	79

It will not always be the case that the Cochran conditions are satisfied for a contingency table with expected values. In that case, the Fisher's Exact Test is used to determine whether there are nonrandom associations between two categorical variables. Fisher's Exact Test calculates a p-value of 0.035 in the case of $cmm_{>1}$ and pmt, leading to the same conclusion as before.

In this section we have analyzed the available data. We have assessed the quality and homogeneity of the data and dependencies between variables. Before we start with our search for risk drivers for misestimation, we explain the basics of logistic regression in the next section.

4. Logistic regression

In this section we dive into the technicalities of IT risk quantification through logistic regression modeling. First, it is worthwhile to give some background on the mathematical methods underlying such analyses.

4.1. Introduction

Logistic regression is a specialized form of regression that is designed to predict and explain a binary categorical variable rather than a metric dependent measure—for instance, what factors or combinations of factors are useful in predicting heart failure, or are significant in explaining buying, or not buying, behavior. Similar in form to regular regression, it can and must be used when the basic assumptions for normal regression, particularly normality of the independent variables, are not met. Usually regression analysis relies on strictly meeting the assumptions of multivariate normality and equal variance–covariance matrices across groups. Logistic regression does not need these strict assumptions.

The binary nature of the dependent variable means that the error term has a binomial distribution instead of a normal distribution, and it thus invalidates all testing based on the assumption of normality. The variance of the dichotomous variable is not constant, creating instances of heteroscedasticity as well. Neither of these can be remedied through ordinary transformations of the dependent or independent variables. Logistic regression was developed to specifically deal with this issue. To that end it makes use of the so called logit transformation, which will be explained further in this section, a special case of what Kendall Atkinson [4] calls "the family of folded power transformations", in which the natural logarithm is used to create "normality by proxy".

Now, compare the quest for IT risk drivers with perinatal epidemiology: one wants to predict neonatal mortality of a child within the first four weeks for live born infants with a birth weight less than 1500 grams, given the gender of the child and the gestational age of the child. In perinatal epidemiology, the dichotomous variable *neonatal mortality* is the outcome variable of interest, and *infant gender* and *gestational age* are the predictors of this outcome. It is relatively easy to know the gestational age or pregnancy time in weeks, and it is trivial to detect the dichotomous gender variable. Usually, some kind of regression analysis based on continuous variables would be used to fit their interrelations, if any. But if the dependent variable, neonatal mortality, can only have two values, this is no longer obvious. One therefore does not model the mean of the outcome variable itself, but the probability that the outcome variable has one of two possible values. This modeling technique is known as logistic regression. In logistic regression you directly estimate the probability of an event occurring. So, one wants to predict the probability of neonatal mortality of children, given their gender and the gestational age. Now, notice that the gender variable has a binary value: boy or girl, and the gestational age variable at birth has a limited range of different values in neonatal mortality: 24 weeks, . . ., 29 weeks. Now we want to turn this discrete input into continuous output, such as a 27.6% chance of mortality for a child with certain indicators.

Because we are just interested in probabilities between zero and one, we have to do a smart transformation to fit these discrete numbers to a value between zero and one. For that, we use the so-called logit transformation, which is crucial for the form of regression that we need for later quantifying IT risks of estimating.

Estimating uses restating probabilities as odds to calculate the logit values. Instead of using ordinary least squares, logistic regression uses the maximum likelihood method by comparing an estimated null model as baseline for a model fit with a proposed model containing the independent variables that are the potential risk drivers.

First, we define the logit transformation and then we will explain its properties.

$$\text{logit}(p) = \log(odds) = \log\left(\frac{p}{1-p}\right). \tag{2}$$

Let p be the probability function of the phenomenon that we are searching for, so it is a function ranging between zero and one. The range of the *odds* of function p is the ratio of the probability of p to that of its alternative $1 - p$, which is $p/(1 - p)$ in a formula. The odds of p ranges between zero and infinity. Now, if we take the logarithm of a function ranging between zero and infinity, we end up with a function that ranges between minus infinity and plus infinity. We do so by taking the logarithm of the odds of p, and hence the above expression, Formula (2). So the logit transforms a range between zero and one into the real numbers. The idea behind this transformation is that if we find some trend with a range we can use the inverse of the logit to bring that range back to the range of the probability function that we are searching for. The trick is thus not to model the probability of a phenomenon itself which potentially predicts values that are theoretically impossible, but to model the logit of the phenomenon, and when a relation is found by statistical means, to convert back to probabilities using the inverse of the logit.

The most well-known modeling technique that is based on this roughly sketched idea is called *logistic regression analysis*. We will apply logistic regression to the data of our case study and use that to model certain IT risks, exactly like how you would model the mortality of prematurely born children.

4.2. The logistic regression model

After having explained the general idea behind logistic regression, we will now explain the necessary mathematics so that we can apply logistic regression modeling in our field study. We explain for the uninitiated how to directly estimate the probability of an event occurring using a logistic modeling technique and how the parameters of the resulting logistic regression equation can be estimated.

Let y_i denote the outcome of the event *cost misestimation* or *no cost misestimation*, which equals cm_i, for project i with the following possible values for the actual costs ac and estimate costs ec:

$$y_i = cm_i = \begin{cases} 1 & \text{if } ac < (ec - 5\%) \text{ or } ac > (ec + 2.5\%) \\ 0 & \text{if } (ec - 5\%) \leq ac \leq (ec + 2.5\%). \end{cases} \tag{3}$$

In our research we used the above boundary values for misestimation, but you are free to choose them otherwise. Let p_i denote the probability of cost misestimation of project i. Let $X_1, X_2, \ldots, X_n$ denote the risk factors on which the outcome of y_i depends. Let β_j denote the impact of risk factor X_j on p_i. So β_j is the weight that can be given to factor X_j, even if its effect is confounded by the presence of other risk factors that influence the outcome of y_i. From each project of our sample we know whether or not cost misestimation occurred and we know the corresponding observations on the risk factors. So we have a sequence of observations at our disposal:

$$\begin{aligned} &(y_1, x_{11}, x_{12}, \ldots, x_{1n}), \\ &(y_2, x_{21}, x_{22}, \ldots, x_{2n}), \\ &\cdots, \\ &(y_m, x_{m1}, x_{m2}, \ldots, x_{mn}). \end{aligned}$$

Here x_{ik} denotes the i-th observation of the explanatory variable X_k. A well-known method for predicting a dependent variable from a set of independent variables is multiple-regression analysis. As stated before, multiple linear regression is not suited for modeling binary data. Let us consider the following equation for multiple linear regression:

$$p_i = \beta_0 + \beta_1 \cdot x_{i1} + \cdots + \beta_n \cdot x_{in} + u_i. \tag{4}$$

In Eq. (4) the disturbance term is denoted by u_i. It is assumed that all the u_i are variables with zero expectation and have equal variance for all i. The explanatory variables are allowed to be all kinds of variables—ordinal, categorical, continuous variables—but the dependent variable needs to be a continuous variable. The coefficients $\beta_0, \beta_1, \ldots, \beta_n$ and the parameters of the distribution of u_i are unknown, and the problem is to obtain estimates of these unknowns. In linear regression the parameters $\beta_0, \beta_1, \ldots, \beta_n$ of the model are estimated using the method of least squares. Doing so in this case, the difficulty is clearly that not all possible values of p_i predicted from the model can be interpreted as probabilities as they are not constrained to fall in the interval between 0 and 1. So, the linear model can predict values which are theoretically impossible: below 0 or above 1. Moreover, the assumptions necessary for testing hypotheses in regression analysis are violated as it is unreasonable to assume the distribution of the errors to be normal if the dependent variable can only have two values. Also, other well-known multivariate statistical techniques such as linear discriminant analysis to predict the membership of a group are also not to be considered, as the assumptions necessary for applying the method are not fulfilled, i.e. multivariate normality of the explanatory variables and equal variance–covariance matrices in the two groups.

Of course, it is also possible to try to create a linear regression model which does not model the chance of project cost misestimation, but the amount of misestimated costs. Such practice is also possible in the perinatal epidemiology case. Modeling the survived number of days of prematurely born infants with a light birth weight, instead of the chance of not surviving, creates a linear regression model as opposed to the common logistic model. But the logistic model, indicating survival of the first 28 days, is of greater interest than the exact number of days that an infant has survived. For reasons similar to those that are apparent in epidemiology, we choose in the underlying case of IT risk also logistic modeling: we

are mostly interested in the presence or absence of cost misestimation, and the factors that influence that outcome rather than the actual amount of misestimation.

Since it is inconvenient to model the probability directly, we apply the logistic regression model. In logistic regression we do not assume that p_i depends on the set of explanatory variables through a linear combination of these variables, but assume that the logistic transformation of p_i, which we denote by $\text{logit}(p_i)$, depends on a linear combination of the explanatory variables, i.e. the risk drivers. This dependency is shown in Eq. (5).

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i2} + \cdots + \beta_n \cdot x_{in}. \tag{5}$$

The logarithms used in this paper are always logarithms with base e, the natural logarithm. Clearly, applying a logarithm to a number is the same as measuring a number on a different scale. Remember that x_{ik} denotes the i-th observation for the explanatory variable X_k with respect to project i.

In matrix notation the logit becomes

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \mathbf{X}_i\beta. \tag{6}$$

In Eq. (6), $\mathbf{X}_i$ denotes the vector of the observations of the explanatory variables with respect to project i: $\mathbf{X}_i = (1, x_{i1} \ldots, x_{in})$, and β is the vector of the coefficients to be estimated: $\beta = (\beta_0, \beta_1, \ldots, \beta_n)$. The choice of the logit transformation of p_i has two advantages. First of all, as stated before, it maps the range [0, 1] onto the range $(-\infty, \infty)$. Second, $p_i/(1-p_i)$ can be interpreted as the *odds* of investment misestimation of project i, which makes a direct interpretation of the regression coefficients possible. A logistic coefficient can be interpreted as the change in the logarithmic value of the odds associated with a one-unit change in the explanatory variable. When we solve p_i from Formula (6) we obtain the logistic regression model:

$$p_i = \frac{e^{\mathbf{X}_i\beta}}{1 + e^{\mathbf{X}_i\beta}}. \tag{7}$$

To obtain estimates of $\beta_0, \beta_1, \ldots, \beta_n$ we use the maximum likelihood method. According to this method we determine the values of p_i for $i = 1, 2, \ldots, n$, which make the observed outcomes $y_1, y_2, \ldots, y_n$ most likely. The y_i are a sequence of zeros and ones. Or, in other words, those estimates β_i are selected that make the observed results p_i as likely as possible.

We assume that the risk of cost misestimation of project i is independent of the risk of cost misestimation of project j for all $i \neq j$. In that case the likelihood function has the form presented in Eq. (8).

$$L(\beta) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}. \tag{8}$$

If we substitute p_i in the right-hand side of Formula (8) with the p_i from Formula (7) and take logarithms of both sides of Formula (8) we obtain the following function of β:

$$\log(L(\beta)) = \sum_{i=1}^{n}[y_i\mathbf{X}_i\beta - \log(1 + e^{\mathbf{X}_i\beta})]. \tag{9}$$

Formula (9) shows the general likelihood function. If we multiply the log-likelihood function by -2 we obtain the so-called deviance of the model by definition. The deviance is just a scaled log-likelihood, and both terms are often used in logistic model evaluation. To find the value of β that maximizes Formula (9), denoted by $\hat{\beta}$, we set the derivative of $\log(L(\beta))$ with respect to β equal to 0 and solve $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_n$ from the resulting system of normal equations. Because the normal equations are nonlinear in the unknowns to be solved, an iterative procedure is applied to obtain the estimates. The estimates can be obtained by the Newton–Raphson procedure if all regularity conditions are fulfilled that are needed to obtain the usual asymptotic properties of the maximum likelihood estimators. We do not go into those mathematical details, and refer the interested reader to the literature on this subject [54,12,35,42]. We assume the regularity conditions to be fulfilled in our case. We have used the logistic regression procedure of the statistical package R [41] to carry out the necessary computations.

The logistic regression procedure of R yields not only $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_n$, but also the standard errors for the estimated parameters. If the sample size is sufficiently large, each regression coefficient is normally distributed by approximation. The standard error can subsequently be used to statistically test whether the estimated regression coefficient is significantly different from 0. If the coefficient does not appear significantly different from 0, using some chosen significance level (e.g. $\alpha = 5\%$), then the corresponding explanatory variable – the risk driver – does not have an influence on the outcome of Y in a statistical sense.

The logistic regression procedure has several methods available for model selection. By model selection we mean that the procedure determines which explanatory variables from the list of available variables, IT risk factors in our case, are meaningful for including into the logistic regression equation. Coefficients are meaningful if they differ significantly from zero, given some chosen significance level (e.g. 5%). Moreover, it should not be possible to increase the likelihood function

significantly by entering one of the non-selected explanatory variables into the regression equation. Basically, there are two approaches for model selection. Both methods are based on the idea of stepwise modeling. One can start with a model that only contains a constant, and at each step the explanatory variable with the highest contribution to the increase of the likelihood function is entered into the model. This method is called the forward stepwise selection method. The backward elimination method starts with entering all explanatory variables on the list into the model. Then, in a number of steps, variables are evaluated for entry or removal. To select variables for removal the likelihood ratio statistic is used.

The deviance of a logistic regression model is defined as -2 times the log-likelihood of that model and has a χ^2 distribution with $N - k$ degrees of freedom, where N equals the number of data points and k the number of parameters in the model [25]. The null hypothesis of the deviance statistic is that the fitted model is not significantly different from a perfect model. A saturated, or perfect, model is a model that explains all variability, or in other words, it contains a binary variable for every data point in the data set. The difference in deviance between two models fitted on the same data set and similar parameters has a χ^2 distribution with k degrees of freedom, with k the difference in number of variables between the two models. Testing for a significant difference in deviance is called the likelihood ratio test and is used to test for inclusion or exclusion of parameters in a model.

For more information on logistic regression we refer the reader to any textbook on the subject, e.g., [25]. A particularly readable introduction to logistic regression with easily understood examples is [10,9]. In the next section we will apply logistic regression to our homogeneous data set.

5. Modeling cost misestimation risks through logistic regression

In this section we show how to find a regression formula for the risk of cost misestimation of an IT-enabled investment project using the logistic modeling technique. Formulas for the risk of schedule misestimates, functionality underdelivery and overall project failure are obtained in a similar way, but not considered in this paper. This article demonstrates the usage of logistic regression in the analysis of IT risks of misestimation with real-world project data with a focus on the risk of cost misestimation.

As in perinatal epidemiology we will model a binary variable indicating whether or not cost misestimation occurred. First, we consider the simplest case, in which there are no risk drivers considered that influence the probability of cost estimation. Following that, we consider the case in which there is one risk driver allowed in the model. And finally, we allow all risk drivers to be considered as variables in the model.

5.1. The case of no risk drivers

If there are no risk drivers that influence the outcome of y_i, then the probability of cost misestimation is equal for all 79 projects that we selected for analysis. In that case Formula (5) boils down to a very simple expression for all projects:

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 \qquad \text{for all } i. \tag{10}$$

Of course we still have to estimate β_0. We do this using the maximum likelihood method. According to this method we determine the value of all p_i that make the observed results $y_1, y_2, \ldots, y_n$, a sequence of ones and zeros, most likely. If we want to maximize the log-likelihood expression, Function (9), for β_0, the following equation represents the model with no risk drivers:

$$\log(L(\beta_0)) = \sum_{i=1}^{n} (y_i \beta_0 - \log(1 + e^{\beta_0})). \tag{11}$$

The latter formula equals:

$$\log(L(\beta_0)) = \beta_0 \cdot \sum_{i=1}^{n} y_i - n \cdot \log(1 + e^{\beta_0}). \tag{12}$$

To find $\hat{\beta}_0$ that maximizes Eq. (12) we take the derivative of Eq. (12), set it equal to zero, and solve the result for β_0.

$$\frac{\partial \log L}{\partial \beta_0} = \sum_{i=1}^{n} y_i - n \cdot \frac{1}{1 + e^{\hat{\beta}_0}} \cdot e^{\hat{\beta}_0} = 0 \tag{13}$$

$$n \cdot \frac{e^{\hat{\beta}_0}}{1 + e^{\hat{\beta}_0}} = \sum_{i=1}^{n} y_i \tag{14}$$

$$\frac{e^{\hat{\beta}_0}}{1 + e^{\hat{\beta}_0}} = \hat{p}_i = \frac{\sum_{i=1}^{n} y_i}{n} = \bar{y} = 0.6962. \tag{15}$$

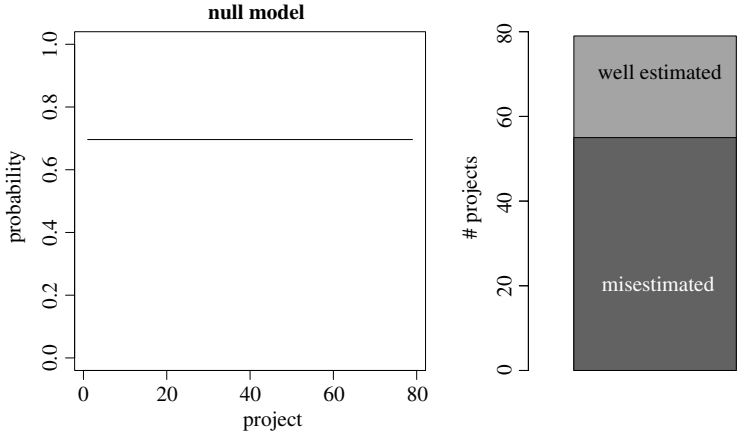

Fig. 4. Plots of the null model and the cost misestimation risk data.

In Eq. (15), $\bar{y}$ stands for the mean of y_i for all i. In that equation we obtain $\hat{p}_i$ directly. Now we can calculate $\hat{\beta}_0$ by using Eq. (7).

$$\Rightarrow \hat{\beta}_0 = 0.8293. \tag{16}$$

So, the simplest estimator of the probability of failure is the number of displayed cost misestimations divided by the total number of projects. In our case study we defined misestimation as more than 5% underrun or more than 2.5% overrun. For our subset of 79 projects we obtain $\bar{y} = 55/79 = 69.62\%$ misestimation, which is a very simple constant model for cost misestimation risk: given a new project, the chance of cost misestimation is 69.62%.

By substituting $p = 55/79$ in Formula (16) we obtain $\hat{\beta}_0 = 0.8293$ as the estimator of β_0 when using the logistic regression model. This is the intercept of a model where the misestimation risk is associated with a constant model, also called the null model. Indeed, if we fit a model using logistic regression with a statistical package we find an intercept of 0.8293, and a log-likelihood of −48.51 and a deviance of 97.02. The deviance of the null model is used as a yardstick to judge whether or not other models that are found later on are an improvement. We recall that when we have found $\hat{\beta}$ we have to transform the model using the inverse of the logitfunction to find the chance of misestimation. But in this simple case of no risk drivers, we already know that value: 69.62%. With Formula (7) we find that the model for cost misestimation risk equals

$$\frac{e^{0.8293}}{1 + e^{0.8293}} = 55/79 = 0.6962. \tag{17}$$

In Fig. 4, we depicted both the model from Eq. (17) and the distribution of the presence and absence of cost misestimation. Indeed, as we can see, the model is just a constant over all the projects. The right-hand side plot is a bar plot showing the frequency of the cost misestimation risk data.

5.2. The case of one risk driver

With the null model we illustrated the basics of applying logistic regression. Let's continue with making the model a little more complex. Suppose we want to know whether conducting a project in-house or outsourcing influences the misestimation risk. To that end we model the misestimation risk for the 79 projects with one categorical variable: whether a project was done in-house or outsourced. This leads to the following logistic regression model:

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 \cdot io_{1i}. \tag{18}$$

In Eq. (18) io_{1i} is a dummy variable that stands for the categorical variable representing in-house projects, $io_{1i} = -1$, or outsourced projects, $io_{1i} = 1$. This so-called Helmert contrast will be explained later. The probability of misestimation now depends on the variable io_{1i}, whether a project was done in-house or outsourced, and therefore is not equal for all projects. If there is just one explanatory variable, the vector β consists of two elements: β_0 and β_1. Using the logistic regression procedure of the statistical package R we obtain Formula (19).

$$\text{logit}(p_i) = 0.746 - 0.235 \cdot io_{1i} \qquad io_{1i} = \begin{cases} -1 & \text{if in-house} \\ 1 & \text{if outsourced.} \end{cases} \tag{19}$$

Therefore, $\hat{\beta}_0 = 0.746$ and $\hat{\beta}_1 = -0.235$. The standard error of the coefficients is 0.260 for both variables.

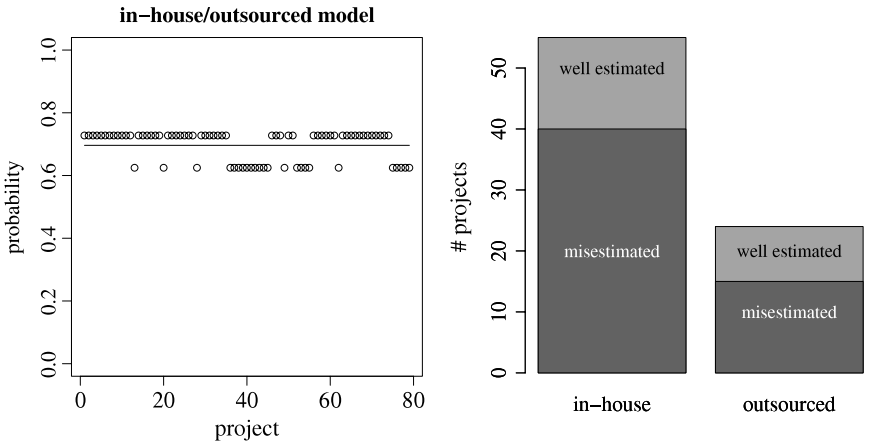

Fig. 5. The in-house and outsourced aware model and the distribution of the misestimation risk.

In Fig. 5 we depict the ensuing model with open circles for the different projects. The left-hand plot contains the new model. As we can see, it is a model with a dichotomous explanatory variable, as is logical given the two possible values for our single variable io_1. To compare the null model with the new model we also depicted the null model with the straight line. The in-house versus outsourced model is a slight improvement over the null model. Indeed, the deviance, −2 times the log-likelihood, is still large, albeit slightly smaller than that of the null model: 96.21, instead of 97.02 for the null model. The purpose of the last model is that if a new project comes in, we predict the cost misestimation risk a little bit more accurately given the extra information of whether the project was done in-house or outsourced. This is only true if this is a significantly better model; we will shortly see that this is not the case. Assuming it is better, then, if a project is going to be outsourced, the risk of cost misestimation changes from

$$p_{cm}(\text{ in-house}) = \frac{e^{0.746-(0.235\cdot-1)}}{1+e^{0.746-(0.235\cdot-1)}} = 0.727$$

to

$$p_{cm}(\text{outsourced}) = \frac{e^{0.746-(0.235\cdot1)}}{1+e^{0.746-(0.235\cdot1)}} = 0.625.$$

Of course, these chances are equal to the proportions of the height of the black boxes in the right-hand side plot of Fig. 5 to the width of the gray boxes. The right-hand side plot shows the distribution of the in-house and outsourced projects, in which the projects with a misestimation are displayed as the black bars.

Note that a statistical analysis indicated that the added value of the variable *io* is not significant. The likelihood ratio test, that we showed in the end of Section 4, for the new model comparing to the null model equals $97.02 - 96.21 = 0.81$. The p-value for this test can be calculated with the χ^2 distribution with k degrees of freedom, with k the difference in number of variables. In this case $k = 1$, which is the difference between a model with only an intercept and a model with an intercept and the variable *io*. This results in a p-value of $P(\chi^2(1) > 0.81) = 0.36$, indicating that we cannot reject the null hypothesis that the coefficient of the variable *io* equals zero. Therefore, we conclude that the coefficient is most probably zero and an insignificant improvement of the model is found, and we will continue our search for a model that is more significant than the null model.

5.3. Creating a full model

Let us now consider the case in which all risk factors that potentially influence the outcome of the event of cost misestimation are taken into account. In Table 11 we summarize the factors that are propagated from the previous Tables 5 and 6. Because we have eliminated two classification factors to create a homogeneous set, these factors are no longer present in Table 11.

In Table 11 we give a summary of the risk drivers and factors that potentially influence the outcome of cost misestimation, *cm*, and that will be used in our further analysis.

Since there are many explanatory variables, and many possible ways to combine them, it is a time-consuming effort to check all the possibilities. In fact, we have ten potential risk drivers, one potential risk factor which cannot be influenced (*pc*) and all potential interactions between each of the risk drivers and the risk factor, so a lot of possibilities to check. This procedure is automated in the statistical package R [41] that we used in this research. The idea is to start with the simplest model, i.e., the null model, and to end up with the model that contains all variables and potential interactions between variables. This occurs if two variables have, besides an individual influence, also a combined influence on the misestimation probability. In the modeling process the step-by-step approach first adds variables, and if necessary later also interaction

Table 11
Generic risk factors at the project level.

Project risk factor	Abbr.	Definition	Level
Estimated costs	*ec*	Size of the project in terms of total estimated costs	Ratio
Estimated duration	*ed*	Size of the project in terms of total estimated duration	Ratio
Project power	*epp*	Estimated size of the project in terms of average costs per month	Ratio
In-house or outsourced	*io*	Project either developed in-house or outsourced	Nominal
IT maturity	$cmm_{>1}$	Reporting unit's CMM level in which the project is done	Ordinal
Reporting unit size	*rs*	Size of reporting unit in terms of total IT costs	Ratio
Project management tool	*pmt*	Project management tool used or not	Nominal
Reporting quality on financial information	*rqf*	Good or not good quality of reporting	Ordinal
Development department size	*dds*	Percentage of development staff in total IT staff	Ratio
Internal development staff size	*ids*	Breakdown of *dds*	Ratio
Project category	*pc*	Transactional, infrastructure, strategic or informational	Nominal

variables. A variable is added if the decrease in deviance of the model is statistically significant. In principle, statistical packages automate the process described, of adding variables, by using the likelihood ratio test based on the difference in deviance of different models. Peduzzi et al. [38] state that at least 10 projects are needed for each explanatory variable in the final model. Since we are adding variables one at a time and have 79 projects, there is no problem for our analysis. Only if we end up with a model with more than eight explanatory variables do we have to reject the resulting model, and remove some of the variables before we start the analysis. The command for the analysis for the statistical package R is the following:

```
step(nullmodel,
    scope=c(lower = ~1, upper = ~.^2 ),
    direction = "both"
)
```

Let us explain the above code snippet, since it illustrates the idea of stepwise modeling. The function `step()` is a generic function that builds models in a stepwise fashion. The first argument `nullmodel` is an object that represents a model of the appropriate class. It does not need to be a good model, as long as it is a feasible model. So, our null model that we depicted in Fig. 4 suffices. The next argument is a `scope` defining the range of models that needs to be examined in the stepwise search for the best model. The notation `lower =~1` means that the crudest model is only having an intercept, which is the null model. The notation `upper =~.^2` means that the most sophisticated model includes all variables, as well as potential interactions between variables in the model. We start with the lower model and step to the upper models until the likelihood ratio test tells us that adding variables is not going to improve the model. We can also start with a full model and remove unnecessary variables. To tell the package to try both directions, forward and backward stepwise modeling, we define the `direction` of the `step()` function to be `both`. The statistical package will then try forward and backward searching and show the best model. Separate forward and backward searches can also be done to check whether the two best models coincide.

Recall that we can step from lower to upper model and vice versa. There is no guarantee that we will end up with the same model. So forward and backward modeling can lead to different models. This is not too strange, since each model is characterized by its deviance, or likelihood, on the one hand, and its complexity in terms of number of variables, on the other hand. Depending on the value of the test statistic, and its corresponding p-value, we decide in a structured way which model is best. A start model that contains variables that are already considered risk drivers can also be helpful in finding the optimal logistic regression model. If one variable shows up in forward and another in backward modeling, it is important to check whether one of the two variables causes the effect of the other variable.

The influence of the different risk drivers is assessed by examining the coefficients of the variables in the various regression equations. The reason for assessing the regression equations is that we can detect both positive and negative influences of certain factors. Such knowledge gives you more grip on the IT function and is an instrument in management based on measured facts. In the following we will discuss different models.

5.4. Helmert treatment

It is not possible to estimate a coefficient for each level of a categorical variable, as the model becomes overparameterized with binary variables for each possible level. Therefore, categorical variables are replaced by sets of binary variables also called dummy variables, as we have already seen in the model with one variable. A particular set of dummy variables is called a set of contrasts. In statistical modeling tools it is possible to use the so-called *Helmert* contrasts to determine the coefficients of the dummy variables used; see Table 12.

The statistical package we use creates by default so-called *treatment* contrasts. In the case of treatment contrasts (see Table 13), one level of each categorical variable is left out and the dummy variable represents the difference between that level and the left-out level. In the case of our categorical variables $cmm_{>1}$, *pmt* and *rqf* we leave out the lowest level, because we are interested in the change of risk when reaching a higher level. This is indeed more in line with our intuition that all projects have at least CMM level 1.

Table 12
Weights of linear combination of the dummy variables according to the Helmert contrasts.

CMM	$cmm_{>1}$	RQF	rqf_1	IO	io_1
CMM 1	−1	no	−1	in-house	−1
CMM > 1	1	yes	1	outsourced	1

PMT	pmt_1	PC	pc_1	pc_2
pmt not used	−1	1	−1	−1
pmt used	1	3	1	−1
		4	0	2

Table 13
Weights of linear combination of the dummy variables according to the treatment contrasts.

CMM	$cmm_{>1}$	RQF	rqf_{yes}	IO	$io_{\text{outsourced}}$
CMM 1	0	no	0	in-house	0
CMM > 1	1	yes	1	outsourced	1

PMT	pmt_{used}	PC	pc_2	pc_3
pmt not used	0	1	0	0
pmt used	1	3	1	0
		4	0	1

Table 14
Confidence intervals of the coefficients of the misestimation risk model [−2.5%, 1.25%].

	Coefficient	95% confidence	90% confidence
Intercept	1.69	(−0.976, 4.356)	(−0.547, 3.927)
dds	4.261	(0.475, 8.046)	(1.084, 7.438)
ids	−3.09	(−5.989, −0.191)	(−5.523, −0.657)

Note that the dummy variable $cmm_{>1} = 1$ stands for CMM level 2 or 3, and that for CMM level 1 the dummy variable has the value 0. This is more natural to interpret and gives us more insight into the influence of going from CMM level 1 to a higher level. We have a similar situation with the variable *rqf* representing a good financial reporting capability and *pmt* representing the usage of a project management tool. For the variable in-house versus outsourced it is not the case that one variable is necessarily better than the other; that is why we use the Helmert treatment for that variable. The most important difference between models with Helmert contrasts and treatment contrasts is the significance of the intercept.

5.5. Search for a model without allowing interaction variables

We now present the model that we found using all data without any interaction variables and applying treatment contrasts to $cmm_{>1}$ and *rqf* in Eq. (20). We used both forward and backward searching.

$$\text{logit}(p_{cm}) = 1.690 + 4.261 \cdot dds - 3.090 \cdot ids. \tag{20}$$

The deviance of this model is 84.488. If we test the difference from the null model, we will see that it is significant: $97.020 - 84.488 = 12.53$. Then, $P(\chi^2(2) > 12.53) = 0.00019$, a significant improvement at the $\alpha = 0.001$ level when comparing to the null model. To apply Formula (20) for other projects, we need to know the values of the variables that are present in the formula. They are *dds*, the percentage of IT staff devoted to development and enhancement activities, and *ids*, the percentage of development and enhancement staff who are internal. With such generic information we can already gain more insight into the misestimation risk of the costs in a project.

As stated, if the sample is sufficiently large, each regression coefficient is normally distributed by approximation. This implies that we can easily assess by inspection whether the value zero falls into the 95% confidence interval of the estimated coefficient. The 95% interval is defined by the mean plus and minus 1.96 times the standard error of the estimated coefficient. The 95% and the 90% confidence interval are shown in Table 14.

The outcome of a logistic regression equation can be turned into a probability by the inverse logit transformation. Let p_{cm} be the chance of cost misestimation risk *cm*. Then Formula (21) shows how the cost misestimation risk is turned into a probability.

$$p_{cm} = \frac{e^{\text{logit}(cm)}}{1 + e^{\text{logit}(cm)}}. \tag{21}$$

The outcome of the logistic regression Eq. (21) is transformed into a risk probability with Eq. (22):

$$p_{cm} = \frac{e^{1.690 + 4.261 \cdot dds - 3.090 \cdot ids}}{1 + e^{1.690 + 4.261 \cdot dds - 3.090 \cdot ids}}. \tag{22}$$

To provide more intuition for this equation, let us calculate the predicted maximum and minimum values of p_{cm}. Our data set contains a project with an *ids* of 0 a *dds* of 0.294. Almost a third of the IT staff were developers, and they were all external. This project yields, using the above formula, a risk of 95% for a disparity between the estimate and its actual. Another project contains an *ids* of 1 and a *dds* of 0.143. 14.3% of the IT staff were developers, and they were all internal.

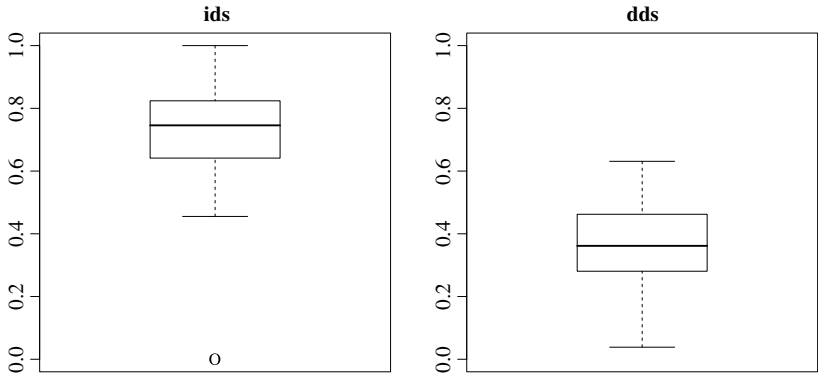

Fig. 6. Box plots for *ids* and *dds* for the data set with 79 projects.

This project predicts a misestimation chance of 31%. As we see, the coefficients of the regression equation are scaled to the values of its corresponding variable. To have a more sophisticated way to assess the quality of the regression coefficients we observe upper and lower bounds of the coefficients provided by the confidence intervals. Each coefficient has a value within these boundaries with 95% and 90% reliability; see Table 14.

The model that we have found contains a constant term and two explanatory variables, risk drivers, of which the regression coefficients differed significantly from zero. The most important risk driver is the percentage of the internal development staff (*ids*) with a negative regression coefficient and the second one is the percentage of development staff in the total IT staff (*dds*) with a positive regression coefficient. Increasing the percentage of internal developers means increasing the specific knowledge within the development team of the core business of the company and its specific culture and this will certainly help in making better estimates. The estimates in this company were all made by internal project managers. This conclusion was supported by the organization that supplied the data set. The internal developers make better estimates as they are much more familiar with testing environments, have better knowledge of the complexity of systems and the infrastructure, and will better judge whether potentially additional requirements are necessary.

Also the regression coefficient of +4.261 of the risk driver *dds* can be interpreted meaningfully. It tells us that efficiency is important for good judgment. An overstaffed development project increases the amount of communication which is often not considered in the cost estimates. Conte et al. [17] discuss the relationship between the size of a team and the productivity which was confirmed by the company that provided the data. With a database of 187 projects they show that the average productivity per person drops when the team size increases. They explain this effect in terms of an increased number of communication paths, citing also the seminal work of Brooks [11]. Brooks states that an increased team size leads to a greater need to coordinate the activities of the group, thus increasing overhead at the expense of production work. As we see in our data set, this effect also influences the quality of estimates. More recent publications on the topic of complex dynamic systems [7,8] further underline this notion of increased and more complex communication in larger development teams.

To give an idea of the common values of the variables *ids* and *dds*, Fig. 6 shows the box plots of these variables for the data set analyzed. Some projects have no internal staff at all, but most projects have development staff for which three quarters are internals. The median value of *dds* is 0.36; the highest value of the ratio of development staff to the total staff is 0.63.

In order to investigate potential interaction between variables we also carried out analyses where interactions of variables are potential risk drivers, besides the risk drivers *ids* and *dds*. This analysis produced the same result as Eq. (20). Apparently, on the basis of our available data, it appears that there is no interaction between the variables *dds* and *ids*.

5.6. Loosening the misestimation risk definition

At this point we have shown the regression equation that we found by applying logistic regression to the available data. We have also shown how to interpret the explanatory variables in the regression equation found. Now, we will consider the stability of the regression equations. Therefore, the bandwidth that indicates budget misestimation is widened. We change the interval from [−5%, +2.5%] to [−10%, +5%]; this results in 48 projects with a cost misestimation. The enlargement of the interval results in fewer cases of misestimation and yields Eq. (23) and the confidence intervals in Table 15.

$$\text{logit}(p_{cm}) = 1.706 - 1.914 \cdot ids. \tag{23}$$

By changing the definition we found a new model that only incorporates *ids* as a significant variable with $\alpha = 0, 1$, besides the intercept. We recall that *ids* is the percentage of internal development and enhancement staff. The effect of increasing the internal staff is the same, a lower risk of misestimation as in the previous model. Although this is a simpler model, we will later see that the goodness of fit of this model is worse than for the previous model containing both *ids* and *dds* based on the smaller interval of good estimates.

Table 15
Confidence intervals of the coefficients of the risk model for misestimation outside [−10%, 5%].

	Coefficient	95% confidence	90% confidence
Intercept	1.706	(0.034, 3.378)	(0.303, 3.109)
ids	−1.914	(−4.111, 0.283)	(−3.758, −0.07)

If a variable definition of the outcome variable, like the misestimation interval, changes, be sure to check that the variable that is being explained does not contain too few true or false cases. For example, if only 2 out of 79 projects are considered to have been misestimated, this will lead to improper conclusions, because the calculations in the logistic regression process need more discriminating data for the outcome variable. A statistical rule of thumb is that we need at least about 20 to 30 values of each possible outcome of the binary variable which needs to be explained.

5.7. Overdispersion

In logistic regression, overdispersion indicates the presence of a larger variability in a data set than is expected from the statistical model underlying logistic regression. Overdispersion is caused by small data sets or small subsets that are induced by different categorical variables, as we have seen before. To prevent overdispersion we removed the categories with a small amount of projects. Also a definition that creates an outcome variable with either very few zeros or very few ones will not lead to a proper analysis. Overdispersion can be detected by dividing the deviance of the logistic model by the degrees of freedom of the logistic model found; this ratio needs to be around 1. If this ratio is improper, larger data sets are needed or a different definition of the variable explained that yields a more discriminating set of trues and falses. Also, a scale factor can be introduced to reduce the variability; this scale factor induces wider confidence intervals, but retains the model found. In our case we have properly mitigated the risk of overdispersion as we have shown in previous sections by taking a subset with enough data points for each classification variable and a proper definition of the outcome variable misestimation.

5.8. CMM insignificant

In the misestimation formula that we found, the influence of the CMM level turned out to be insignificant. We suspect that the skills needed for estimation are independent of the CMM level which is geared towards process maturity. Our data set contains a dichotomous variable that yields 0 in the case of CMM level 1, and in the case of CMM level 2 or 3 the variable has the value 1. This turned out to be a statistically insignificant variable, which was recognized by the company that supplied the data set, which we explain below. In Section 7, we will see that also digging deeper in the data set does not reveal an influence of the CMM level on the correct estimation of costs of an IT-intensive project.

At first sight, this outcome contradicts the definitions of the second level of the CMM model. Level 2 of the CMM model [37], labeled *Repeatable*, dictates that: "It is characteristic of processes at this level that some processes are repeatable, possibly with consistent results". But it also states that "there could still be a significant risk of exceeding cost and time estimates". So, correct estimation is desirable at level 2 and higher, but in the underlying case, in which CMM audits were conducted for the various reporting units, the higher CMM levels did not significantly improve the cost estimation practice. According to the definition of CMM level 2 a probability of exceeding costs still exists. Apparently this probability of cost misestimation is factual for a considerate amount of the projects in our data set. The CMM audits were done by the organization itself and not by a certified third party. Therefore the qualifications of CMM level 2 are possibly optimistic. But it is more likely that the absence of improved estimates is explained by the following. If a higher CMM level is reached, the requirements process within the IT department improves, but the business department falls behind. For instance, the business does not correctly consider testing phase efforts, leading to incorrect estimates, despite the higher CMM level. This is more widely known as a maturity mismatch [14]. A discrepancy in maturity level between the business and IT or supplier and client nullifies the expected benefits of the higher CMM level.

5.9. Pooling in-house and outsourced

In this section we return to the issue of pooling the in-house and outsourced projects. The data set that we have been analyzing did show significant variables, *ids* and *dds*, with an acceptable goodness of fit. But it did not show an influence of the variable *io*, whether or not a project is outsourced. This result justifies the pooling of the data.

The same holds for the categorical variables *pmt*, *rqf* and $cmm_{>1}$. Since these variables did not show up in the logistic regression equation there was no specific influence of one of the separate values of these variables.

6. Goodness of fit and predictive power

We have inferred several cost misestimation models, and now we will assess their goodness of fit and predictive power. Note that you cannot tacitly assume predictive value with such models, since they are built for analytical purposes. We need to assess the goodness of fit first.

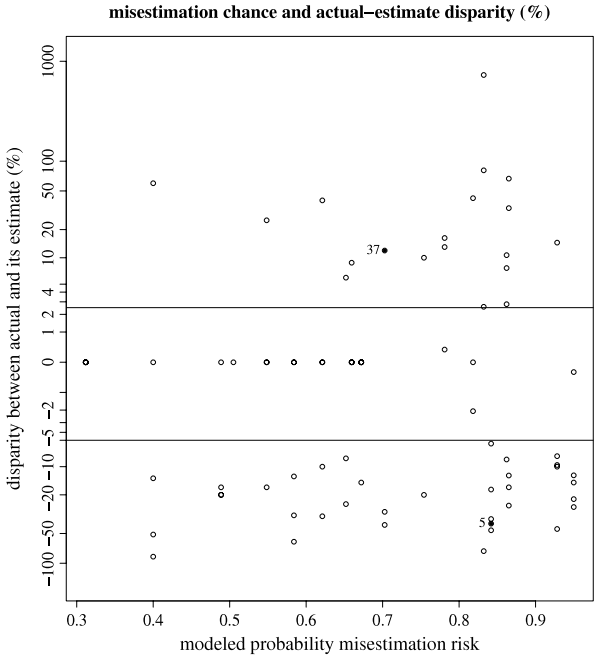

Fig. 7. Quality plots of individual risk probabilities of the cost misestimation risk model [−5%, 2.5%].

6.1. Goodness of fit

In the following sections we start with the assessment of the goodness of fit of the models found. We will inspect the graphical goodness of fit as well as the calculated goodness of fit.

6.1.1. Graphical goodness of fit inspection

We will use the cost misestimation risk model with the interval [−5%, 2.5%] as an example to illustrate two graphical quality plots. In the next figure we illustrate the calculated probabilities of misestimation of the model.

Fig. 7 depicts the risk probabilities of cost misestimation versus the actual misestimations, measured after completion of the projects. To obtain more spread on the vertical axis, the disparity percentages are placed on a logarithmic scale. Therefore, the projects that are close to the misestimation interval of [−2.5%, 5%] are more visible. The vertical lines represent the interval that defines the boundaries of good estimates.

It is not possible to count exactly 79 dots in Fig. 7, because some projects have equal values for the predicted chance of misestimation and the actual misestimation and therefore appear as one dot. Since the explanatory variables in our model are business unit specific variables, Fig. 7 displays vertical lines of projects with equal chances, but different actuals.

We highlighted some projects in the figure, for instance project number 5. These projects realized a large saving in the estimated investment. The model predicted high chances, higher than 80%, of cost misestimation, which is a very good prediction of the model. Upon inspection of the project data, it turned out that project 5 was a combined project where software and business processes were reengineered. The high predicted risk of cost misestimation is in line with the risks of large business/software reengineering projects. The actual 40% cost savings of this project were due to proper risk mitigation efforts by the organization. Project 37 on the other hand displays an overrun of costs. From the plot it seems that the larger the chance of cost misestimation, the more likely it is that there will indeed be a misestimation. But when the probability is low it seems that there is less predictive power. To make this more precise, we will plot the chance that a cost misestimation occurs given a minimal predictive value from our model.

For instance, if the model calculates a chance of between 30% and 40% of misestimation, what is the chance that an actual misestimation occurs? We measure the amount of actual misestimations that indeed emerged and took the fraction of the total number of projects where our model predicts between 30% and 40% chance of cost misestimation. In this case, there are nine projects with that chance and four of them have an actual misestimation. So, when we predict a chance of 30%–40% of misestimation, there is a 44% chance that there is going to be an actual misestimation. In Fig. 8 we depict this relation for all decimals. Indeed for higher predicted misestimation chances, the chance that an actual misestimation will occur increases. In the next section we calculate the goodness of fit more formally.

6.1.2. Calculating goodness of fit metrics

With the deviance of our models we tested whether the misestimation models do not perform significantly worse than a perfect, saturated model. If we perform this likelihood ratio test [25] for the [−5%, 2.5%] null model, the p-value equals

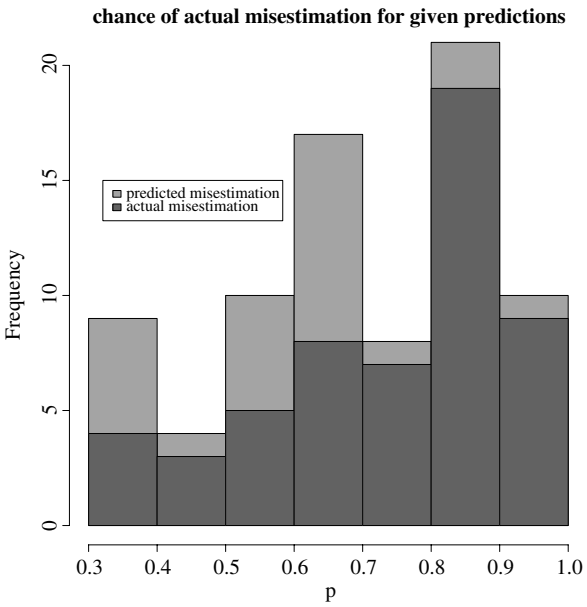

Fig. 8. The chance of actual misestimation given a prediction from the cost misestimation model.

Table 16
A confusion matrix for a two-class case.

Prediction	Actual	
	Misestimate	Correct estimate
Misestimate	True positive	False positive
Correct estimate	False negative	True negative

0.07, and we need to reject the hypothesis that the model performs as well as a perfect, saturated, model. The p-value for this test for our [−5%, 2.5%] model, with *ids* and *dds* as risk drivers, is as follows: 0.236. In this case, we cannot reject the null hypothesis for any common α level, and conclude that we have a model that does not perform significantly worse than a perfect model. For the model with the [−10%, 5%] interval we do reject this null hypothesis for the null model and the fitted model with *ids* as a risk driver.

As in linear regression, there exist goodness of fit metrics for logistic regression [26]. The test of the unweighted sum of squares has a p-value of 0.98 for the [−5%, 2.5%] model, indicating a good fit. The Hosmer–Lemeshow goodness of fit statistic $\hat{C}$ is calculated by ordering all projects by their predicted chance of misestimation and subsequently partitioning the projects over k groups, usually $k = 10$ [25], for which n_k indicates the number of projects in group k. For each partition the number of actual misestimations, o_k, and the number of expected misestimations, e_k, based on the chances in group k, are calculated. The statistic $\hat{C}$ is then calculated as follows:

$$\hat{C} = \sum_{k=1}^{10} \frac{(o_k - e_k)^2}{e_k \cdot \left(1 - \frac{e_k}{n_k}\right)}. \tag{24}$$

The Hosmer–Lemeshow goodness of fit test compares the observed and estimated expected frequencies for the k groups. The value of $\hat{C}$ for the [−5%, 2.5%] misestimation model equals 10.11246, it has a χ^2 distribution with eight, $k - 2$, degrees of freedom and therefore a p-value of 0.257, indicating a well fitted model.

For the [−10%, 5%] model the $\hat{C}$-statistic has a value of 6.83, with a p-value of 0.555; in other words: both are well fitted models.

6.1.3. Banana lifting

A tabular way to evaluate a model's goodness of fit for classification purposes consists of simply classifying successes and errors of the model. We distinguish four kinds of successes and errors: true positives and false positives, and true negatives and false negatives. We can put this in a classification matrix, also called a confusion matrix [55,28].

Table 16 illustrates the idea. True positives are projects that had an actual misestimation and are also predicted to have a misestimation. False positives are projects that have no significant disparity between the estimate and the actual, but are predicted to have a misestimation. The other half of the table can be read in a similar way. Now we must choose a probability for distinguishing a risky from a non-risky project. This probability enables us to provide values for the confusion

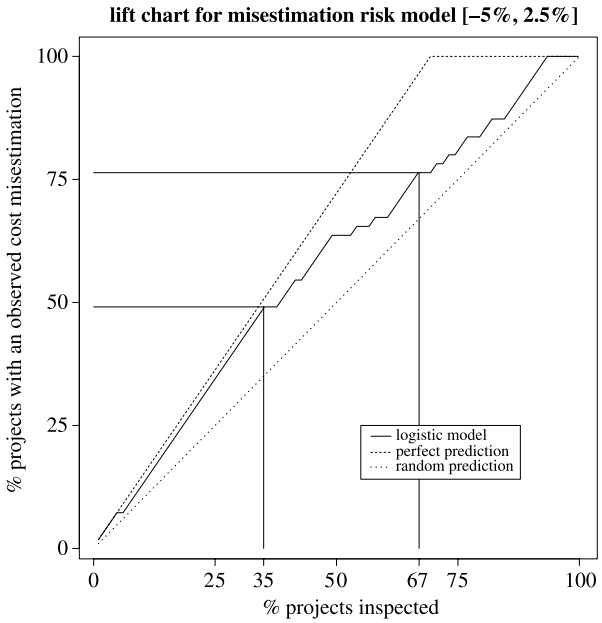

Fig. 9. Lift chart for the cost misestimation model.

matrix. The crux is how to choose this probability. An ideal graphical support technique for selecting this probability is the *lift chart*, sometimes called a *banana chart* because of its visual appearance [55,28]. The lift chart shows how a response variable behaves when a prediction model is used. The chart displays three lines, a baseline that represents a random choice of projects, a perfect prediction line and the lift curve itself induced by the model. The lift curve displays, one hopes, an increase in response rate, which is called the lift. A lift chart indicates which subset of the data contains the largest possible proportion of responses, in our case misestimations. The further the lift curve is away from the baseline, the better the performance of the model. So, in fact the lift shows how much better the model is than a random pick. To create a lift chart, instances in a data set are sorted in descending order of their probability of a misestimation. Plotting the sorted data creates a graphical depiction of the various probabilities. A lift chart is thus ideal for giving an overview of the classification power of a model.

Fig. 9 depicts a lift chart for our cost misestimation model for the data set of 79 projects. The horizontal axis represents the inspected projects, ranked by decreasing predicted misestimation risk as predicted by our misestimation risk model. On the vertical axis we ranked the projects with an observed misestimation. The solid staircase-shaped curve represents the lift chart of our predictive model. This line is formed by sorting the predicted risks from high to low and comparing them with the observed misestimation, which is whether a project did suffer from a significant disparity between the estimate and its actual or not. Each time an actual misestimation is detected, the line is lifted.

The dashed line with an angle of 45° is the predictive power of a random prediction: detecting 50% of the projects with misestimated costs is achieved by inspecting 50% of the projects at random. The random model is just the one where all projects have a risk probability of 50%. The dashed line segments above the prediction represent the perfect prediction: every project is predicted correctly. This means that the 55 projects inspected first are all the projects with a cost misestimation. This line turns horizontal after the first 55 projects inspected. Our lift chart is enclosed within the region created by a random and a perfect pick: it is better than random, but not as good as a perfect prediction. The ideal place to be in the lift chart is near the upper left-hand corner: the speed for detecting risks is optimal, since all actual risky projects are detected first due to the ordering by misestimation chance. The lift chart of the cost misestimation model for the interval [−5%, 2.5%] on the overall data set is reasonable. It is well above the line of random predictions for half of the data set, but overall not much better than a random pick. This is not strange, because the model is based on the same data set as it is predicting the chances for. Later we will apply this method in simulation of random subsets of our data sets.

The leftmost labeled point in the lift chart indicates that inspecting 35% of the projects ranked according to our model leads to a detection of 49% of the projects with an actual misestimation. Our model thus started reasonably with a lift of 49% when inspecting 35% of the projects, yielding a lift factor of $49/35 = 1.4$. From that point on the lift factor diminishes: the line moves away from the ideal prediction. The next labeled point in the chart has a lift factor of only 1.13 (76% found by inspecting 67% of the projects). The lift chart tells us that if the risks of misestimation are lower, the predictive power of the model is also somewhat lower. The lift factor of a random prediction can be calculated as follows: the last inspected project is the last misestimated project to be found. This equals a lift factor of $\frac{100\%}{100\%} = 1$. If after an inspection of 50% of the projects only 25% of the misestimates are found, the lift factor for this subset equals $25/50 = 0.5$.

Table 17
A confusion matrix when 25% of the projects are inspected according to our cost misestimation risk model [−5%, 2.5%].

Predictions	Actuals		
	Misestimates	Good estimates	Totals
Misestimates	24	1	25
Good estimates	31	23	54
Totals	55	24	79

Table 18
Average lift factors of the misestimation risk.

	Average lift factor
Risk for misestimation outside [−5%, 2.5%]	1.148
Risk for misestimation outside [−10%, 5%]	1.088

We noted that ratios like 49/35 were instructive for assessing the quality of our lift chart. To make this more formal, we introduce definitions taken from [55]. They are the *recall* and the *precision* of a lift chart. The recall is the ratio of the true positives, the predicted misestimates that are also actual misestimates in our case, divided by the sum of the true positives and the false negatives. The precision is the ratio of the true positives divided by the sum of true and false positives. We calculate the recall and precision for a lift chart by inspecting the confusion matrix. Suppose that we inspected 25% of the projects ordered with a decreasing cost misestimation risk; then Table 17 gives us the correct amount of true and false positives and negatives.

The recall at 25 inspected projects is $24/(24+31) = 0.45$, or 45%, with a corresponding precision at 25 inspected projects of $24/(24+1) = 0.96$. A summary measure for recall and precision is the so-called three-point average precision or the eleven-point average precision. This is the average precision at certain recall levels. For the three-point summary this is the average precision at 20%, 50% and 80%, and for the eleven-point average precision this is the average precision at 0%, 10%, . . . , 100%. The summary measures for this model are 84.75% and 75.10% respectively. So, the average precision of our cost misestimation model for interval [−5%, 2.5%] is about 80%. We expected to detect $55/79 = 0.70$ or 70% of the projects with a cost misestimation, which implies an average lift factor of about 1.14 (80/70) in the lift chart. This average lift factor can be seen as an overall quality of lift charts. These are relative measures and are used to compare between different models for the same risk. We have computed the average lift factors on the basis of the earlier defined subsets in order to determine the most stable classification model. We depict these lift factors for the cost misestimation risk model in Table 18.

From Table 18 we observe that our misestimation models have average lift factors of 1.15 and 1.09. Our cost misestimation risk model performs in a stable manner with respect to the classification performance of the data set on which the model was built. The lift charts and metrics used so far indicate that our misestimation risk model classifies the projects decently, but it does not help us in determining the ideal amount of projects to control. To that end, we calculate the so-called F-measure that represents the information retrieval quality of the model [55]. This measure is defined as follows:

$$F\text{-measure} = \frac{2 \cdot recall \cdot precision}{recall + precision}.$$

We compute the F-measure for each possible inspection amount of projects with the above formula. In Fig. 10 we plot the number of projects inspected against their corresponding F-measures. A high risk detection quality is expressed by high values of the F-measure.

Fig. 10 shows us that we need to inspect at least 40 projects to detect most actual misestimated projects; after 30 the F-measure still rises but not as fast as before the first 40 projects inspected.

With the F-measure we were able to obtain an indication of the ideal control amount of projects based on the risk model for cost misestimation. This indication is based on theoretical measures and past performance since the models were assessed with the historical data on which the models were built.

6.2. Predictive power of the cost misestimation risk models

Having a model does not imply that you can use it for anything you like. Suppose we want to use the model to predict misestimation risks of individual projects; we have to judge its predictive power first, which we will do in the following. In this section we compare the predictive quality of the models found earlier on the basis of the estimation intervals [−5%, 2.5%] and [−10%, 5%]. The predictive quality of a model is measured by the so-called Mean Minus Log-Likelihood (*MML*), which calculates the deviation of the predictions from the original response variable [27,12]. If the original data set is used to calculate the *MML*, it equals minus the log-likelihood divided by the size of the this data set, which explains its name: minus the average, or mean, of the log-likelihood. The *MML* is presented in Formula (25).

$$MML = -\frac{1}{N} \cdot \sum_i \left[Y_i \log \hat{p}_{(-i)}(X_i) + (1 - Y_i) \log(1 - \hat{p}_{(-i)}(X_i)) \right]. \tag{25}$$

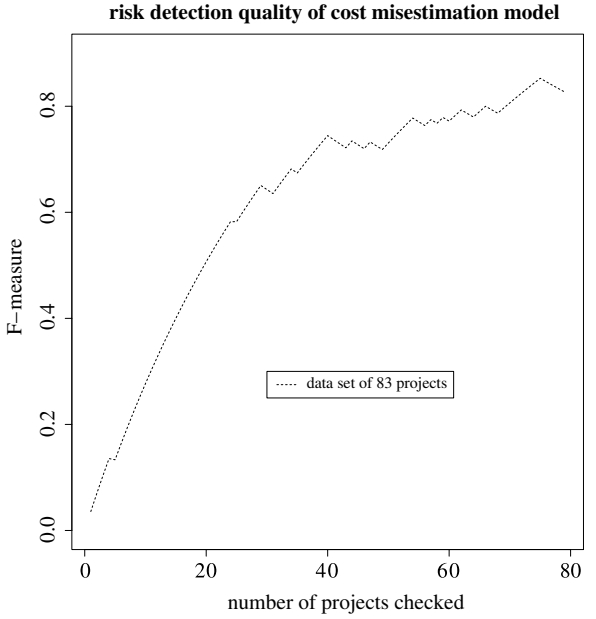

Fig. 10. *F*-measure for each possible control amount of projects for the cost misestimation risk model.

Table 19
Performance of overall models for the different subsets.

	MML
[−5%, 2.5%]	0.588
[−10%, 5%]	0.742

In Formula (25), $p_{(-i)}(X_i)$ stands for the predicted chance of misestimation for project i based on a misestimation model created with all data except project i; hence the subscripted minus i in brackets. The perfect prediction of a data set has an *MML* of zero. The worst value is $+\infty$. We consider a model as having an acceptable predictive power if its *MML* is between zero and 0.5.

Zooming in on the models (Table 19), we notice that the cost misestimation model with the small interval is performing better than the model with the larger interval. The model with the [−5%, 2.5%] interval has an *MML* that is almost acceptable. A larger data set is needed to draw more definite conclusions on the predictive value of the model on the individual project level. Given the available data set though, it is possible to better understand the predictive value of the models. We investigate this in more detail in the following sections.

Note that the calculation of the *MML* through Formula (25) is needed if no test set is available and we resort to the original data. If a test set of projects is available those projects are used to calculate a mean minus log-likelihood. In that case the logistic regression model found on the basis of the research data set is used to predict the chance of the misestimation of the project costs of the test set.

6.2.1. Simulation

Since the *MML* did not provide enough evidence of the predictive power for classification purposes for our model, we run a simulation. In this simulation we create random subdivisions of our 79-project data set. Note that the ideal way to assess the predictive power of a model is to use it to predict the risk of cost misestimation for a larger number of new projects and to evaluate the successes and errors. However, such a set is often not available. In that case a simulation as described in this section is an alternative. With a random subset with a size of 80% of the 79 projects we create a cost misestimation model. With this model based on about 63 projects we calculate the average of the three- and eleven-point lift factors of the remaining 20% of the data set, i.e. 16 projects. This process is repeated 10,000 times. From the 10,000 lift factors we calculate the probability density function, which is displayed in Fig. 11. In this way we have simulated the situation in which the research data set for constructing the model consists of 63 projects and the test set consists of 16 projects.

In Fig. 11 we have plotted the lift factor of the original model as well as the median lift factor of the randomly subdivided sets. Given a random subdivision of the data set of 79 projects, a similar lift factor is to be expected. Since we had no separate test set for testing the predictive power of our [−5%, 2.5%] model, we created random partitions to check whether the lift factor of the [−5%, 2.5%] model is an exceptional lift factor or not. As Fig. 11 shows, the lift factor based on the entire research

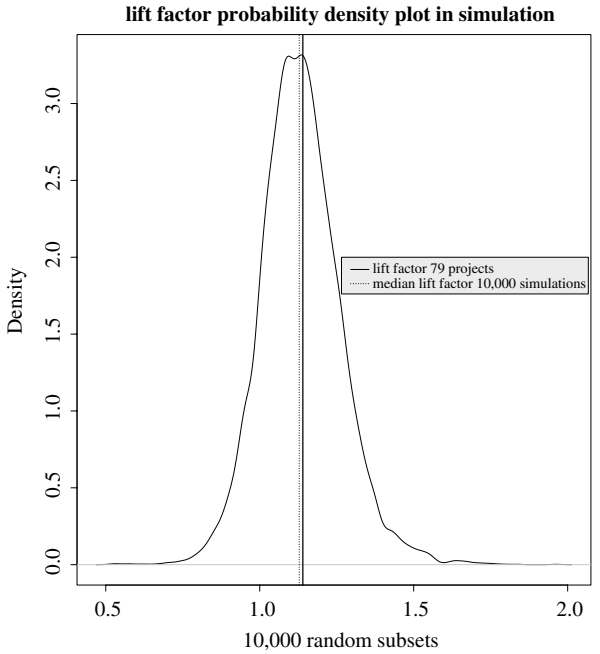

Fig. 11. Distribution of lift factors obtained from 10,000 simulations of random subset division for which the 80% subset served to create a model and the 20% complement to calculate the lift factor.

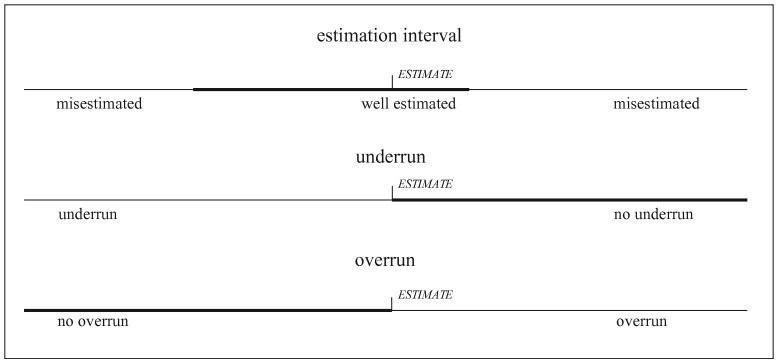

Fig. 12. Illustration of different risk materialization definitions.

set is almost equal to the median of the 10,000 simulations. Apparently, the earlier calculated lift factor of 1.15 already gave a good indication of the predictive power of the model.

At this point we are in a position to conclude that our models are useful for finding the risk drivers of misestimation, because of the goodness of fit. The predictive power of the model for individual projects is rather poor as the *MML* displays, but for the classification of portfolios of projects it is useful as we have seen with the lift models.

7. Separating misestimation into overrun and underrun

In Section 5 we have seen that the explanatory variables for misestimation are *dds*, the ratio of enhancement and development staff to the complete IT staff, and *ids*, the ratio of internal development and enhancement staff. Now we will dig deeper into the data set and study the influences on overrun and underrun separately, which we previously combined into a misestimation that incorporates both overrun and underrun. In Fig. 12 the differences of the definition of the risk materialization are illustrated.

The first part of Fig. 12 illustrates the model of the misestimation already examined. In that case an actual value that is outside predefined boundaries of the estimate, indicated by the thick line, counts as a misestimation. The second part of Fig. 12 represents the underrun situation: an actual that is located on the thin line counts as an underrun. Overrun is defined by an actual value that is located on the thin line of the third part of Fig. 12. In all three definitions the comparison of an actual with an estimate results in a dichotomous variable fit to be used in logistic regression. The first case of logistic regression,

using an interval, was extensively explained in the previous sections. As we have our data set up for logistic analyses in the previous sections, it is close to effortless to analyze alternative scenarios.

First, we examine the alternative of underrun. Recall that underrun is defined as an actual that turned out to be lower than the estimation. Using the same techniques as we elaborately discussed in this paper, we obtain the following equation by applying logistic regression for the chance of underrun, denoted by p_{ur}:

$$\text{logit}(p_{\text{ur}}) = 2.148 + 0.714 \cdot rqf_{\text{yes}} - 3.531 \cdot ids. \tag{26}$$

The standard errors for the variables in Eq. (26) are respectively 0.948, 0.502 and 1.301. The data set for the underrun model contains 38 misestimations. Second, we calculate the logistic regression equation for the chance of overrun, denoted as p_{or}. Overrun is present if an actual value is higher than the estimate. This model contains 20 misestimations. The resulting equation from the logistic regression method is as follows.

$$\text{logit}(p_{\text{or}}) = -2.879 - 1.202 \cdot rqf_{\text{yes}} + 6.111 \cdot dds. \tag{27}$$

The standard errors for the variables in Eq. (27) are respectively 0.868, 0.664 and 2.403. What strikes one about Eqs. (26) and (27) is that the variables *ids* and *dds* each only appear in *one* equation. But they have the same effect on the misestimation if an interval is considered, as was elaborately discussed in Section 5 with Eq. (20). Increasing the ratio of development staff increases the risk of overrun and the risk of misestimation, when considering an interval. Increasing the ratio of internal staff decreases the risk of underrun and the risk of misestimation if an interval is considered. Apparently if the overrun and underrun effects are combined, the separate causes *ids* and *dds* remain risk drivers in the resulting logistic regression equation.

Another striking point of Eqs. (26) and (27) is the appearance of the variable rqf_{yes}. This variable has the value 1 if financial reporting is practised and 0 if this is not done. In the equation for underrun there is a higher chance of underrun if financial reporting is present. In the equation for overrun on the other hand, its presence displays a decrease in the chance of overrun. So, if financial reporting is practised, there is both a decrease in the chance of overrun and an increase in the chance of underrun. Since no other variables regarding for instance the costs or duration appear in these equations, the two samples are of a similar nature as regards size. These equations support the notion that the reporting variable is not visible in the equation for misestimation if an interval around the estimate is inspected. Apparently, if financial reporting is conducted, projects do not become compliant to target, but drop significantly below their estimate, and become underruns below the boundaries of the inspected interval. Estimates for projects with financial reporting tend to be too high, where their opposites tend to be too low. If no financial reporting is conducted, financial estimates tend to be risk seeking, and if the reporting is conducted, the estimates tend to be risk mitigating.

8. Conclusions

In this paper we have shown that logistic regression is a powerful modeling technique for investigating what risk factors influence the risk of a significant disparity between a cost estimate and its actual, or, for short, misestimations. The definition of *significant* in this context is defined by the bounds of an interval around the estimate. In this paper we mainly researched an interval of $[-5\%, +2, 5\%]$ around the estimate; an actual that is situated outside this interval is considered to be misestimated. If the project costs or schedule are incorrectly estimated, wrong decisions are made at the start of the project as regards allocation of money and staff capacity. Moreover, calculations of the Return on Investment (ROI), the Net Present Value (NPV) and Pay Back Time (PBT) are based on wrong figures and this may lead to the acceptance of unsound investment proposals. It therefore makes sense to analyze your set of IT projects and investigate for risk drivers that cause misestimations.

We have focused on the risk of falsely estimating the costs, being one of the most critical KPIs of an investment project. The authors are comfortable that the method developed for cost misestimation is applicable for investigating the significant risk drivers for project schedule misestimation and the risk of functionality underdelivery if the necessary data are available. However, we have not carried out that exercise, as the focus of the organization that provided the data is on cost management.

For our case study we found that the differences between the development environments of the reporting units explain the variety in the chances of project cost misestimation. According to the logistic regression equation found, the risk of misestimation varies within the range of 0.31 to 0.95 for the various reporting units. The chance of misestimation is independent of project specific characteristics. It only depends on the characteristics of the development environment. To be more precise, it varies with the percentage of developers in the total IT staff, the metric *dds*, and the percentage of the development staff who are internal, the metric *ids*. For management this information is extremely useful. It tells management that the focus needs to be on *dds* and *ids* in order to improve the estimation quality. Of course the coefficients of the regression equations that we have found are specific to the data set that we have researched, but the risk drivers found are more generally applicable.

Our most important learning experiences are listed below.

- It is very important to inspect the data carefully before applying the logistic modeling technique. It does not make any sense to use an entire data set of projects if the set is not homogeneous. In that case a regression equation will be found that does not apply to any of the homogeneous subsets of which the total data set consists. In our case we detected one subset of 79 projects, out of a total set consisting of 165 projects, which satisfied the condition of being homogeneous with respect to the research questions under consideration and consisting of sufficient data.
- The regression equation found must have a logical interpretation. In our case we found a constant term and two risk drivers for which the regression coefficients differed significantly from zero. The most important risk driver was the percentage of the internal development staff, *ids*, and turned out to have a negative regression coefficient, and the second one was the percentage of development staff in the total IT staff (*dds*) with a positive regression coefficient. The question is whether one can put a meaningful interpretation on the risk drivers found and the signs of their regression coefficients. In our case the answer is *yes*, and the statistical conclusions were supported by the company providing the data set. Increasing the percentage of internal developers means increasing the specific knowledge within the development team of the core business of the company and its specific culture, complexity, infrastructure and requirements process. Apparently this aids in making better estimates. Also the regression coefficient of $+4.448$ of the risk driver *dds* can be interpreted meaningfully. This tells us that efficiency is important for good judgment. Projects that are overstaffed increase the amount of communication paths which is often not considered in the cost estimates. This results in a misestimation of the project costs. It is remarkable that the CMM level turned out not to be significant in the regression equations as one of the risk drivers. A significant improvement of estimations is expected on the basis of the definitions of the capability maturity model. The CMM audits were done by the company itself, and not by certified CMM auditors, making the level 2 qualifications probably optimistic. But the absence of improved estimates in higher CMM levels is more likely induced by a maturity mismatch. The higher CMM level improves requirements processes, but the business is not aware of efforts needed for testing, leading to incorrect estimates. The variable that indicates reporting on financial information turned out to have a decreasing effect on the risk of estimate overrun, and also an increasing effect on the risk of estimate underrun. When considering the more general notions of misestimation, both underrun and overrun, the effects of financial reporting cancelled each other out for the separate cases. The underlying systematics are probably best described as follows. In reporting units without reporting, projects tend to be risk seeking, and in reporting units with reporting, they are more risk avoiding.
- The model has been very useful for identifying the important and less important risk drivers in the collected data as the goodness of fit showed. To our surprise only two risk drivers turned out to have a significant influence on the estimation quality. For management this is valuable and useful information. It tells management that it must first of all focus on these two risk drivers to improve the estimation quality and management can neglect the other potential risk factors for the time being, for instance an effort to go from CMM level 1 to CMM level 2 to improve the estimation practice.
- The predictive power of the model for individual projects is not acceptable as the *MML* displays, but for the classification of portfolios of projects it is useful, as we have seen with the lift models. To reduce the cost of auditing, our models provide nonrandom selections of projects that have the highest chance of misestimation problems. This aids in focusing attention first on the most risky projects in terms of the largest chance of misestimated project costs or project duration.

We stress that the coefficients of the formulas presented in this paper cannot be used verbatim by other organizations and are specific to the organization providing the data. However, the risk

drivers found are in our opinion points of interest and attention in IT governance in other organizations, especially those in the financial services industry. The conclusions presented are useful as guidelines even in the absence of the data necessary for constructing logistic regression models. Moreover, the methods explained in this paper for identifying the drivers for IT risks are universally applicable for obtaining your own IT risk models.

Capturing misestimation risks implies, among other things, that given a set of easily retrieved indicators as regards an IT project and the environment in which it is carried out, a prediction can be made as to how large the IT risks will be, so that proper measures can be taken to mitigate them.

Acknowledgments

This research received partial support from the Dutch *Joint Academic and Commercial Quality Research & Development (Jacquard)* program on Software Engineering Research via contract 638.004.405 *Equity: Exploring Quantifiable Information Technology Yields* and via contract 638.003.611 *Symbiosis: Synergy of managing business–IT-alignment, IT-sourcing and offshoring success in society*. We would like to express our gratitude to Joeri van Hoeve who laid the foundation for the studies presented in this paper. Furthermore, we would like to thank the anonymous reviewers for their valuable comments.

References

[1] A.J. Albrecht, Measuring application development productivity, in: Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium, 1979, pp. 83–92.
[2] A.J. Albrecht, J.E. Gaffney, Software function, source lines of code,and development effort prediction: A software science validation, IEEE Transactions on Software Engineering 9 (9) (1983) 639–648.

[3] G.M. Allenby, P.J. Lenk, Modeling household purchase behavior with logistic normal regression, Journal of the American Statistical Association 89 (1994).
[4] Kendall Atkinson, Elementary Numerical Analysis, John Wiley & Sons, 1985.
[5] Steven Bellman, Gerald L. Lohse, Eric J. Johnson, Predictors of online buying behavior, Communications of the ACM 42 (12) (1999) 32–38.
[6] S. Berinato, Using project portfolio management to demonstrate IT value, CIO Magazine, October 2001. Available via: www.cio.com/archive/100101/math.html.
[7] Dan Braha, Yaneer Bar-Yam, Topology of large-scale engineering problem-solving networks, Physical Review E 69 (1) (2004).
[8] Dan Braha, Yaneer Bar-Yam, The statistical mechanics of complex product development: Empirical and analytical results, Management Science 53 (7) (2007) 1127–1145.
[9] R. Brand, Using logistic regression in perinatal epidemiology: An introduction for clinical researchers. Part 2: The logistic regression equation, Paediatric and Perinatal Epidemiology 4 (1990) 234–249.
[10] R. Brand, M.J.N.C. Keirse, Using logistic regression in perinatal epidemiology: An introduction for clinical researchers. Part 1: Basic concepts, Paediatric and Perinatal Epidemiology 4 (1990) 22–38.
[11] F.P. Brooks Jr., The Mythical Man-Month — Essays on Software Engineering, Anniversary Edition, Addison-Wesley, 1995.
[12] S. le Cessie, Model building techniques for logistic regression, with applications to medical data. Ph.D. Thesis, University of Leiden, 1991.
[13] Jason Charvat, Project Management Methodologies, Wiley, 2003.
[14] Timothy A. Chick, CMM/CMMI level 3 or higher? No guarantee for success. Defense AT&L, November–December 2006.
[15] William G. Cochran, The χ^2 test of goodness of fit, The Annals of Mathematical Statistics 23 (3) (1952) 315–345.
[16] William G. Cochran, Some methods for strengthening the common χ^2 tests, Biometrics 10 (4) (1954) 417–451.
[17] S.D. Conte, H.E. Dunsmore, V.Y. Shen, Software Engineering Metrics and Models, Benjamin-Cummings Publishing Co., Inc., 1986.
[18] D. Dalcher, A. Genus, Introduction: Avoiding IS/IT implementation failure, Technology Analysis and Strategic Management 15 (4) (2003) 403–407.
[19] G. Denaro, M. Pezzè, S. Morasca, Towards industrially relevant fault-proneness models, International Journal of Software Engineering and Knowledge Engineering 13 (4) (2003) 395–417.
[20] J.L. Eveleens, C. Verhoef, Quantifying IT forecast quality. Accepted with minor revision, available via: http://www.cs.vu.nl/~x/cone/cone.pdf.
[21] D. Garmus, D. Herron, Function Point Analysis – Measurement Practices for Successful Software Projects, Addison-Wesley, 2001.
[22] The Standish Group, CHAOS, 1995.
[23] The Standish Group, CHAOS: A Recipe for Success, 1999.
[24] The Standish Group, EXTREME CHAOS, 2001.
[25] D. Hosmer, S. Lemeshow, Applied logistic regression, in: Probability and Statistics, 2nd edition, John Wiley & Sons, Inc., 2000.
[26] D.W. Hosmer, T. Hosmer, S. Le Cessie, S. Lemeshow, A comparison of goodness-of-fit test for the logistic regression model, Statistics in Medicine 16 (1997) 965–980.
[27] J.C. van Houwelingen, S. le Cessie, Predictive value of statistical models, Statistics in Medicine 9 (11) (1990) 1303–1325.
[28] Joyce Jackson, Data mining: A conceptual overview, Communications of the Association for Information Systems 8 (19) (2002).
[29] C. Jones, Patterns of Software Systems Failure and Success, International Thomsom Computer Press, 1996.
[30] C. Jones, Software Assessments, Benchmarks, and Best Practices, in: Information Technology Series, Addison-Wesley, 2000.
[31] Taghi M. Khoshgoftaar, Edward B. Allena, Wendell D. Jones, John P. Hudepohl, Accuracy of software quality models over multiple releases, Annals of Software Engineering 9 (1–4) (2000) 103–116.
[32] G.P. Kulk, C. Verhoef, Quantifying requirements volatility effects, Science of Computer Programming 72 (3) (2008) 136–175. Available via: http://www.cs.vu.nl/~x/qrv/qrv.pdf.
[33] Rensis Likert, A technique for the measurement of attitudes, Archives of Psychology 140 (1932) 1–55.
[34] F.W. McFarlan, Portfolio approach to information systems, Harvard Business Review 59 (5) (1981) 142–150.
[35] I. Newton, Methodus fluxionum et serierum infinitarum, Londoni, 1671. English translation by John Colson 1736.
[36] Mukta Paliwala, Usha A. Kumar, Neural networks and statistical techniques: A review of applications, Expert Systems with Applications 36 (2009) 2–17.
[37] Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber, Capability maturity model, version 1.1, IEEE Software 10 (4) (1993) 18–27.
[38] P. Peduzzi, J. Concato, E. Kemper, T.R. Holford, A.R. Feinstein, A simulation study of the number of events per variable in logistic regression analysis, Journal of Clinical Epidemiology 49 (1996) 1372–1379.
[39] R.J. Peters, C. Verhoef, Quantifying the yield of risk-bearing IT-portfolios, Science of Computer Programming 71 (2007) 17–56. Available via: http://www.cs.vu.nl/~x/yield/yield.pdf.
[40] J. Drew Procaccino, June M. Verner, Scott P. Overmyer, Marvin E. Darter, Case study: Factors for early prediction of software development success, Information and Software Technology 44 (1) (2002) 53–62.
[41] R Development Core Team, R: A Programming Environment for Data Analysis and Graphics. R Foundation for Statistical Computing, Vienna, Austria, 2.7.1 - 2008. ISBN 3-900051-12-7.
[42] Joseph Raphson, Analysis aequationum universalis, Londini, 1690.
[43] R. Schmidt, K. Lyytinen, M. Keil, P. Cule, Identifying software project risks: An international Delphi study, Journal of Management Information Systems 17 (4) (2001) 5–36.
[44] N.F. Schneidewind, Investigation of logistic regression as a discriminant of software quality, in: Seventh International Symposium on Software Metrics, IEEE Computer Society, 2001, pp. 328–337.
[45] Adam Smith, An Inquiry into the Nature and Causes of the Wealth of Nations, in: S.M. Soares (Ed.), MetaLibri Digital Library, vol. 1776, 2007.
[46] C. Verhoef, Getting on top of IT, 2002. Available via: http://www.cs.vu.nl/~x/top/top.pdf.
[47] C. Verhoef, Quantitative IT portfolio management, Science of Computer Programming 45 (1) (2002) 1–96. Available via: http://www.cs.vu.nl/~x/ipm/ipm.pdf.
[48] C. Verhoef, Quantifying software process improvement, 2004. Available via: http://www.cs.vu.nl/~x/spi/spi.pdf.
[49] C. Verhoef, Quantifying the value of IT-investments, Science of Computer Programming 56 (3) (2005) 315–342. Available via: http://www.cs.vu.nl/~x/val/val.pdf.
[50] C. Verhoef, Quantitative aspects of outsourcing deals, Science of Computer Programming 56 (3) (2005) 275–313. Available via: http://www.cs.vu.nl/~x/out/out.pdf.
[51] C. Verhoef, Quantifying the effects of IT-governance rules, Science of Computer Programming 67 (2–3) (2007) 247–277. Available via: http://www.cs.vu.nl/~x/gov/gov.pdf.
[52] J.M. Verner, W.M. Evanco, N. Cerpa, State of the practice: An exploratory analysis of schedule estimation and software project success prediction, Information and Software Technology 49 (2) (2007) 181–193.
[53] P. Weill, M. Broadbent, Leveraging the New Infrastructure, Harvard Business School Press, 1998.
[54] Eric W. Weisstein, Newton's method. Available at: http://mathworld.wolfram.com/NewtonsMethod.html.
[55] Ian H. Witten, Eibe Frank, Data Mining, Practical Machine Learning Tools and Techniques, Elsevier, 2000.
[56] G. Peter Zhang, Mark Keil, Arun Rai, Joan Mann, Predicting information technology project escalation: A neural network approach, Computing, Artificial Intelligence and Information Technology 146 (1) (2003) 115–129.