Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information and Software Technology 48 (2006) 302–310
INFORMATION AND SOFTWARE TECHNOLOGY
www.elsevier.com/locate/infsof

Software effort estimation terminology: The tower of Babel

Stein Grimstad*, Magne Jørgensen, Kjetil Moløkken-Østvold

Simula Research Laboratory, P.O. Box 134, NO 1325 Lysaker, Norway

Received 5 November 2004; revised 15 April 2005; accepted 19 April 2005
Available online 13 June 2005

Abstract

It is well documented that the software industry suffers from frequent cost overruns. A contributing factor is, we believe, the imprecise estimation terminology in use. A lack of clarity and precision in the use of estimation terms reduces the interpretability of estimation accuracy results, makes the communication of estimates difficult, and lowers the learning possibilities. This paper reports on a structured review of typical software effort estimation terminology in software engineering textbooks and software estimation research papers. The review provides evidence that the term 'effort estimate' is frequently used without sufficient clarification of its meaning, and that estimation accuracy is often evaluated without ensuring that the estimated and the actual effort are comparable. Guidelines are suggested on how to reduce this lack of clarity and precision in terminology.

Keywords: Software effort estimation; Terminology; Structured review; Software estimation guidelines

1. Introduction

Software development effort estimates are the basis for project bidding, budgeting and planning. These are critical practices in the software industry, because poor budgeting and planning often has dramatic consequences. When budgets and plans are too pessimistic, business opportunities can be lost, while over-optimism may be followed by significant losses. The importance of accurate estimates is documented in a wide range of studies. For instance, the Standish Group [24] concludes that reliable estimation is among the top ten most important success factors in software projects. It is therefore unfortunate that, as indicated in a recent review of estimation surveys [41], there has been little improvement in software cost estimation accuracy over the last 20 years. We believe that one reason for this lack of improvement is the imprecise use of terminology for effort estimation. The following two case stories indicate that proper communication, interpretation and improvement of estimation accuracy measurements may be a problem when there is no precise use of terms related to estimation. This problem motivates the review and guidelines provided in this paper.

Case story 1: In 2003, two of the authors performed a survey on project estimation in Norwegian software companies [43]. The goal was to obtain an in-depth understanding of estimation practice and to examine factors that affect the accuracy of effort estimation. The basis for the measurement of estimation accuracy was a comparison of the actual use of effort with the estimated most likely effort provided in the planning stage of the project, i.e. the amount of effort the contractor believes that the project will require, regardless of the price to the customer or the budget. An interesting result was the observation that government projects had, on average, significantly higher deviations between estimated most likely efforts and actual efforts than private projects [42]. This observation made the headlines in Norway's largest morning newspaper. The day after the results were presented, the front page of the newspaper stated 'Yearly overruns of 6 billions [Norwegian Kroner] in governmental IT-projects'[18]. The debate that followed was heated, and culminated in the research results being discussed in the Norwegian parliament. In particular, there were members of parliament who saw our results as evidence of a waste of government money on IT projects. However, our results did not say anything about the *customers*' budget overruns or losses. Neither had we studied the software providers' budget overruns or losses.

* Corresponding author. Tel.: +47 67 82 82 00; fax: +47 67 82 82 01.
E-mail addresses: steingr@simula.no (S. Grimstad), magnej@simula.no (M. Jørgensen), kjetilmo@simula.no (K. Moløkken-Østvold).

0950-5849/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2005.04.004

What we *did* study was the overruns related to what the software providers believed was the *most likely effort* of a project. The newspaper article, which was the basis for the debate, did not point this out, i.e. we had failed to communicate the difference between overruns of most likely effort and overruns of budgeted costs. Budgeted costs typically include a risk buffer added to the most likely effort. The cost overrun we found through the survey was therefore probably much higher than the software organizations' and the customers' budget overrun. A consequence of the misinterpretation of the term 'cost estimate' was that the public discussion focused mainly on whether one should believe the high cost overrun number or not, and much less on how government projects could be better managed, i.e. on improvement of their role as software customers.

Case story 2: Some time ago, one of the authors was hired as a solution architect of a software project. It was a high risk project for a number of reasons: the functionality to be developed was complex, several stakeholders with conflicting goals were involved and a non-extendable deadline was set. The initial analysis suggested that the project would involve about 40 people and changes had to be made to five systems, all in operation. Our schedule and effort estimates suggested that we could probably deliver before the deadline, but with small margins. Not surprisingly, we ran into trouble during development and the changes to one of the systems were two weeks delayed. The changes to this system were on the project's critical path and the entire project was therefore two weeks delayed. Moving the deadline was, of course, unacceptable to the customer because this would ruin the announced launch. However, we did manage to deliver all functionality on time and on budget. We did so in the same way that many other software development teams do in similar situations; we reduced the amount of testing. The project went into operation, and luckily only minor failures occurred. How accurate were our estimates? From the outside, i.e. as would have been observed in most estimation surveys, we had only minor effort estimation error and no schedule overrun. In reality, however, the project would have had greater estimation error and a time overrun if the testing process had been completed as planned, i.e. with the promised level of quality. This case story shows that common ways of measuring effort estimation accuracy may give a misleading picture of the real estimation accuracy and hence a misleading picture of the need for improvement in the process of estimation and project management.

We present related work in Section 2 of this paper. Section 3 further elaborates on the consequences of imprecise use of effort estimation terminology. The consequences are illustrated by estimation error analyses of software projects in a Norwegian software development organization. In Section 4 we review the actual use of effort estimation terminology in popular software engineering textbooks and software estimation research papers. Based on the related work in Section 2, the discussion in Section 3 and the review in Section 4 we provide, in Section 5, recommendations aimed at improving the use of software effort estimation terminology, thus enabling improvement in the process of estimation. Section 6 concludes.

2. Related work

The problems of imprecise terminology for software cost estimation have been addressed by several software engineering researchers. Table 1 lists a number of research papers and textbooks that suggest there may be problems caused by the imprecise use of software estimation terminology. However, we have been unable to find any study with the same goal as this paper, i.e. to documenting the importance and the severity of the terminology problem through a structured review of popular software engineering textbooks and a representative set of estimation research papers.

3. The importance of precise effort estimation terminology

Obviously, the use of more precise or standardized terminology is not sufficient to solve most of the problems inherent in effort estimation. However, we believe that a *necessary* condition for sustainable improvement is the precise use of important terms, because lack of precision leads easily to the following:

- A mix of processes with different purposes, e.g. a mix of processes with a focus on realism (estimation of most likely effort), a focus on efficient development work (estimation of planned effort), a focus on avoidance of budget overrun (decisions on budgeted effort), and a focus on winning a bid (estimation of price-to-win). The lack of separation between these processes have been found to reduce the realism of estimation of most likely effort [10,12,32].
- Comparison of estimation error of different projects when they are not really comparable. For example, one project may have based their estimation error measurement on the difference between planned effort and actual effort, while another project may have based theirs on the difference between most likely effort and actual effort.
- Survey results that are difficult to interpret. If you want to evaluate your own estimation performance against those presented in an estimation survey, this is virtually impossible if the survey results are based on an unknown mixture of estimates of different types and different sizes of contingency buffers.

The consequences are improper evaluation, comparison and reporting of effort estimation performance, including a lower ability to learn from experience [26,30].

Table 1
Related work

Author	Examples of estimation terminology problems addressed
Kitchenham [33]	Kitchenham recommends that before you improve estimation processes, you should make sure that you do not have a management problem. She identifies lack of understanding of the probabilistic nature of estimates; confusing plans, costs and estimates; and not giving sufficient time to the estimation process as specific management problems
DeMarco and Lister [14]	The authors argue that schedule flaws can occur when no distinction is made between the most optimistic estimate (that with virtually no probability of success), the goal (that which the project aims for), the estimate (the most likely outcome) and the schedule (what the project commits to)
DeMarco [13]	DeMarco points out that an estimate is a prediction based on a probabilistic assessment, and that an estimate should be the most likely value accompanied by upper and lower bounds. He says lack of experience is one of the important reasons for poor estimation and proposes that a separate metrics group should be responsible for data collection and estimation
Boehm and Fairley [7]	Boehm and Fairly state two important points about software estimation: (1) It is best to understand the background of an estimate before using it and, (2) It is best to orientate the estimation approach to the use that is going to be made of the estimate
Edwards and Moores [15]	The authors show that estimates can be a rough guide to the cost of a project as well as applying numbers to the detailed project plan. These meanings of estimates are different with respect to uncertainty, usage and motivation. They argue that the lack of clear distinction between these two types of estimates is why estimation tools are not commonly used in the industry
Coombs [11]	Coombs explains that mixing price, cost and realistic estimates together with reduced functionality gives a false impression of estimation accuracy. He claims that this happens because most projects are underestimated to begin with and the only option left for the project managers is to axe the requirements and use both the contingency allowance and profit in order to meet the budget
Jørgensen [25]	The author explains, through comparison with vacation cost estimation and by an industrial case study, why cost estimates with similar accuracy can hide huge differences in estimation performance. The paper exemplifies the conflicting goals of different types of estimates; 'most-likely software development cost', 'risk-minded planned development cost' and 'cost-reducing planned development costs'

Recent observations of software projects in a Norwegian software development organization further illustrate these consequences. Over a period of two years we logged estimation information of several software projects in that company as part of a study on reasons for estimation errors (a subset of the data is presented in [29]). As a part of the logging we requested that the person responsible for the estimation documented the estimate of 'most likely effort'. An analysis of the description of how the estimate of 'most likely effort' was derived showed, however, a wide variety of interpretations. In most cases, the estimate was *not* of most likely effort. Instead, the effort estimate was typically described as most likely effort plus a risk buffer of varying size, i.e. it was interpreted as the planned or the budgeted effort, or was sometimes described as the effort (derived from the price) agreed with the customer. A consequence of the imprecise use of 'effort estimate' in the studied organization was that it was difficult to compare and evaluate the estimation accuracy of different projects. We compared the subset of projects that we assessed to contain effort estimates of 'most likely effort' with those we assessed to contain effort estimates of a type used as a basis for price to customer or planned effort, i.e. where a risk buffer typically was added to the most likely effort. The remaining projects were omitted from the comparison because it was difficult to classify the types of estimate used. The estimates of most likely effort had, on average, an effort overrun of 11%, while the estimates including a risk buffer on average used 8% less effort than estimated. From the description of the estimation process it seems that a typical risk buffer was 10–20% of most likely effort. When removing the specified risk buffer from the estimates that contained them, we found that they had, on average, almost the same estimation accuracy (about 10% overrun) as the estimates described as 'most likely effort'. Finding the average estimation accuracy of all projects, without adjustments, would be like adding 'apples and oranges'.

In the studied organization we also found it necessary to adjust the actual effort for the decreases and increases in delivered functionality to enable a proper interpretation. Several of the projects had increases or decreases in functionality of more than 10%. In most cases this adjustment led to better estimation accuracy; many estimates of most likely effort were accurate, but looked inaccurate because of added or removed functionality. For example, a project went from 50% effort overrun to 10% overrun when we adjusted for the increase in functionality. Again, without this adjustment a comparison of estimation accuracy would give a misleading picture of the estimation ability in many of the projects.

We have logged similar estimation information for other organizations and believe that this is a common pattern [28]. The results are further supported by the survey described in the first section of this paper [43], which indicates, for instance, that the separation between price and most likely effort is blurred in many organizations.

Lack of precise terminology for estimation may hinder the improvement of estimation processes. It is, for example,

difficult to determine whether estimation errors are caused by poor estimation ability, poor risk analysis, poor project management, or something else, when a clear estimation terminology is absent.

4. A review of textbooks and previous research

4.1. Design of review process

In order to examine possible reasons for the lack of precise estimation terminology in the software industry, we reviewed the actual use of estimation terminology in a representative set of estimation research papers and the most popular software engineering textbooks. The motivation for this selection is, that we believe these are the publications that have the most influence on the estimation terminology used by software professionals. The review focuses on the following two questions derived from the software cost estimation terminology problems discussed in the previous sections:

- Q1: Is the term 'effort estimate' precisely defined?
- Q2: When evaluating estimation accuracy, are the estimates and the actual efforts comparable?

In order to conduct the review in a fair and auditable manner, the review design is based on the guidelines for systematic reviews proposed by Kitchenham [34]. Note that the review design deviates somewhat from these guidelines. The main reason for the deviations is that we aim at describing 'typical practices', not 'best practices' or 'all practices', regarding use of estimation terminology. Section 4.2 describes how the reviewed material was selected, while Section 4.3 explains how the review questions (Q1 and Q2) were assessed for the reviewed material. The results of the review are presented in Section 4.4. The validity of the review is discussed in Section 4.5, and Section 4.6 provides a general discussion of the results.

4.2. Selection of reviewed material

We used different approaches to select textbooks and research papers. As we aimed at selecting textbooks that address software cost estimation and are among those most often read by software professionals, we targeted books used by universities in lecturing and books that software professionals read. For research papers, we consider precise estimation terminology to be most important for papers that report on estimation accuracy. Therefore, we aimed at selecting a representative sample of research papers in journals and conferences that report on software cost estimation accuracy.

Three different information sources were used to identify the literature. Lecture books were found by using Google to search the internet using the search string 'software engineering course books'. We then manually investigated the first 100 URLs that appeared relevant, and counted the frequency of each textbook used in university courses on software engineering. The four most popular textbooks were included in the review.

Books that software professionals are most likely to read were selected by using Amazon's list of the top 100 bestselling computer science books. Books were extracted from the bestselling list by browsing the list manually. For a book to be included in the review, we had to regard it as likely that the book addresses software cost estimation in particular. These judgments were based on the title and the abstract of the book. The assessments were done by two of the authors, independently of each other. When there were disagreements, the books were included. If the same author had multiple books that appeared to meet the selection criteria, only the most recent were selected. An examination of the books revealed that three of the books did not include any sections on software cost estimation. These three books were excluded from the review.

Research papers were selected from the BESTweb library (available at www.simula.no/BESTweb). BESTweb is an online library of estimation papers that claims to include nearly all journal papers and a large proportion of the conference papers on software cost estimation. The journal papers in BESTweb were selected by manually scanning potentially relevant journals, while conference papers were identified by a comprehensive search in the INSPEC-library. A full description of the BESTweb library will appear in a forthcoming paper by one of the authors. At the time of the review, the BESTweb library contained 963 estimation relevant articles. We selected papers to include in the review by reading all the abstracts, and then the full versions of all the papers that appeared to meet all inclusion criteria:

- Deal with estimates of software development effort, schedule, budget or cost or with project success/failure/performance.
- Report on empirical collected estimates from real projects (not experiments or student projects).
- Report on estimates made up-front. This excludes, for example, all history-based evaluations of formal estimation models.
- Report on estimation accuracy.
- The paper was the most recent paper by the main author that met the above criteria. This criterion applied to several of the authors (for instance, Lederer and Prasad had several older papers excluded).

This selection procedure has some limitations, e.g. it is not a review of all relevant estimation papers and the initial selection of papers by title and abstract only may lead to exclusion of papers that otherwise meet the inclusion criteria. For the purpose of documenting typical estimation

terminology practice, however, the selection process is, in our opinion, acceptable.

The searches for lecture books and bestselling books were conducted on the 23rd of February 2005. Amazon's best selling lists report those books that currently have the most sales. Similarly, most of the web pages of university courses in software engineering were recently updated. The search for research papers was conducted on the 19th of February. BESTweb covers articles published up to April 2004. The selection of reviewed lecture books and research articles were done by one of the authors, while two of the authors selected bestselling books.

4.3. Assessment of the research questions (data extraction)

The first review question (Q1: Is the term 'effort estimate' precisely defined?) is evaluated to have been answered satisfactorily if there is a definition of 'effort estimate' in the reviewed material that clarifies whether the intended meaning of the term is an estimate of 'most likely effort', 'budgeted effort', 'price' or something else. The question is also deemed to have been answered satisfactorily if, even if there is no explicit definition, the terminology in use makes a clear and consistent distinction between estimates made for different purposes.

The second review question (Q2: When evaluating estimation accuracy, are the estimates and the actual efforts comparable?) is evaluated to have been answered satisfactorily if the comparability of estimated effort and actual effort is discussed, or actions to ensure comparability are taken in situations when there may be significant differences in functionality or quality between the estimated and the actual solution. For example, comparable values can be secured either through adjustments of actual effort or removal of projects in cases where estimated and actual effort is not comparable.

Assessment of the research questions was conducted by two of the authors, independently of each other. When a question was not a topic addressed in a textbook or a paper, we used the value 'na' (not addressed). There were only minor disagreements to be resolved. Disagreements were resolved by examining each controversial issue separately. We reached an agreement in all the cases.

4.4. Review results

The reviewed material and our evaluations of their use of estimation terminology are presented in Table 2 (textbooks) and Table 3 (research papers). In the tables, the first column identifies the reviewed publication, the second and third columns report the type of publication and the fourth and fifth columns include our answers of the review questions (Q1 and Q2).

The results of the review show that the term 'effort estimate' is rarely used in a consistent manner (Q1) neither in textbooks nor research papers. Only one [39] out of the eight books and two [27,35] out of the 23 research papers use estimation related terminology in a way we find satisfying according to the criteria described in Section 4.3. However, a few of the texts partly comply to the criteria. For instance, in [22] there is a distinction between estimates for pricing and estimates for planning, but no distinction between 'planned effort' and 'most likely effort'. For the majority of the reviewed material we were unable to tell whether the term 'effort estimate' referred to an estimate of 'most likely effort', 'budgeted effort', 'price' or something else.

Whether estimated effort is comparable to actual effort when evaluating estimation accuracy (Q2) is a topic in two of the textbooks [21,39]. They both include a brief discussion of problems related to estimation accuracy evaluation, but neither of them suggest guidelines or provide any example of how the problems can be solved in practice. In the research papers, Q2 is addressed in nine of

Table 2
Textbooks

Publication	Lecture book	Bestselling book	Q1	Q2
Brooks [9]	Y	Y	N	NA
Heldman [21]	N	Y	N	Y
Larman [36]	Y	N	N	NA
McConnell [39]	N	Y	Y	Y
Pressman [44]	Y	Y	N	NA
Schwaber and Beedle [46]	N	Y	N	NA
Sommerville [48]	Y	N	N	NA
Sponsky [49]	N	Y	N	NA

Table 3
Research papers

Publication	Conference paper	Journal paper	Q1	Q2
Abdel-Hamid et al. [1]	N	Y	N	Y
Barki et al. [3]	N	Y	N	N
Barry et al. [4]	N	Y	N	Y
Bergeron and St-Arnaud [5]	N	Y	N	Y
Berry and Schoenborn [6]	Y	N	N	Y
Bootsma [8]	Y	N	N	N
Fleck [16]	N	Y	N	N
Gray et al. [17]	Y	N	N	N
Haynes and Henderson-Sellers [19]	N	Y	N	N
Heemstra and Kusters [20]	N	Y	N	N
Hill et al. [22]	N	Y	N	Y
Jenkins et al. [23]	N	Y	N	N
Jørgensen [27]	N	Y	Y	N
Kamatar and Hayes [31]	N	Y	N	N
Kitchenham et al. [35]	N	Y	Y	Y
Lederer and Prasad [37]	N	Y	N	N
Lind and Sulek [38]	N	Y	N	N
Mizuno et al. [40]	Y	N	N	Y
Moløkken and Jørgensen [41]	Y	N	N	N
Ropponen and Lyytinen [45]	N	Y	N	N
Shepperd and Cartwright [47]	N	Y	N	Y
Subramanian and Breslawski [50]	N	Y	N	N
Taff et al. [51]	N	Y	N	Y

the papers [1,4–6,22,35,40,47,51]. They handle the incomparability in a somewhat different manners: Some studies discuss the consequence of incomparability or assess it to be ignorable/not relevant [1,4,5,40,47], some studies remove data points [22,35], while one study avoids to calculate estimation accuracy at all due to comparison problems [51]. Only one of the studies attempts to adjust the actual effort to be comparable to the original estimate [6].

4.5. Threats to validity

We assess the major threats to validity to be related to: (1) Biased selection of textbooks or research papers, (2) Biased review of the textbooks or research papers.

4.5.1. Biased selection

The bestselling list at Amazon and the first 100 hits on Internet is a sample based on current popularity and not, for example, quality. This type of sample was intended as a means to review typical practice, but it does give a poor picture of 'best practice' among textbooks. We acknowledge that there are textbooks with rather precise estimation terminology. Another limitation to the review is that it only includes books and articles that address software cost estimation. This means that related material, such as general project management and forecasting literature, where a more precise terminology might be present, was not reviewed. However, our impression is that such literature is not much read by most software professionals.

4.5.2. Biased review

The review was conducted by two reviewers, independently of each other. However, these reviewers are from the same research group and so they are not totally independent, and other reviewers may answer the questions differently. In addition, some of the reviews may be highly subjective. Despite these potential sources of bias, we believe that the main conclusion is quite robust: most textbooks and research papers on estimation are not based on a precise use of estimation terminology.

4.6. Discussion of results

The previous sections suggest that an important obstacle for estimation improvement is imprecise estimation terminology and that a reason for the lack of precise use of estimation terminology is the lack of precise terminology in software textbooks and research papers. However, it might also be that the direction of cause and effect is reversed, as well. It is difficult to survey estimation practice and write good estimation guidelines when important estimation terms are vague and used inconsistently by software professionals. Consequently, attempts to improve the use of estimation terminology should be made concurrently in both industry practice and the writing of textbooks and research papers.

There may be different motivations for improving the precision of estimate terminology. Software organizations may wish to improve their use of estimation terminology to avoid misunderstandings, to increase the realism in the estimates, and to facilitate learning from experience. Software researchers may wish to develop precise terminology to increase the validity of their research results, (e.g. when comparing two formal estimation models), and to suggest better estimation guidelines for software professionals.

Since several researchers have pointed out the importance of a precise software estimation terminology, e.g. DeMarco [13] as early as in 1982, it is somewhat surprising that software estimation has been conducted out for so many years without greater attention being paid to the use of precise terminology. There are a number of possible reasons for this lack of progress:

- Authors of the estimation literature seem to take a 'deterministic' (estimates as one single effort value) instead of a 'probabilistic' (estimates as a combination of effort value and probability) view on effort estimation. A probabilistic view means here that 'most likely effort', 'planned effort', 'budgeted effort', etc., are values (with different probabilities of being exceeded by actual effort) on an effort probability distribution. Fig. 1 illustrates how estimates of most likely effort (the effort with the highest probability) and planned effort (typically, most likely effort plus a contingency allowance) are values on a probability distribution of effort. Without a probabilistic basis of effort estimation terminology a separation of most likely, planned, and budgeted effort may be difficult to describe. The strong textbook focus on parametric cost estimation models, which typically deliver only a single effort value, may be one reason for the adoption of a deterministic view.
- Software organizations do not regard estimation as a separate activity, but as an integrated part of project planning, project pricing and project budgeting. As pointed out earlier, mixing processes may result in the mixing of terminology.

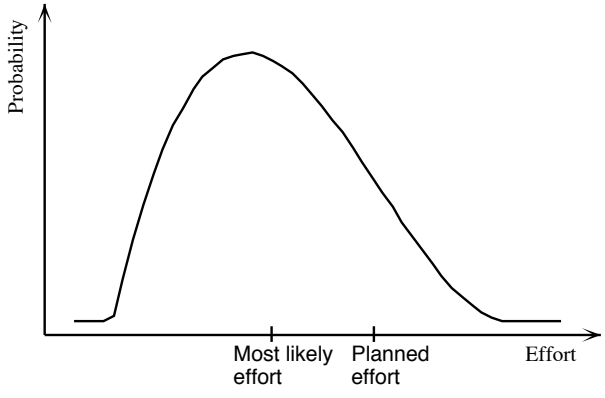

Fig. 1. Example of an effort probability distribution.

- Software organizations typically do not collect the data necessary to validate and adjust the actual effort to make it comparable with the estimated effort. Our experience is that most organizations have an unformed view on how to assess estimation accuracy measurements and do not allocate any resources to the in-depth analysis of estimation accuracy data across projects [30].

5. Guidelines for estimation terminology

Our review motivates a change towards a more proper software cost estimation terminology among software professionals and researchers. Proper estimation terminology is a complex topic and it is beyond the scope of this paper to provide suggestions for a complete terminology. We propose, however, two simple guidelines, the following of which is, we believe, essential for improved software estimation processes. The guidelines are aimed at all users of software cost estimation terminology, including authors, practitioners, researchers and reviewers. The guidelines are based on our own experience and recommendations made in the text books and papers summarized in Table 1.

Guideline 1: Do not mix estimation of most likely effort with planning, budgeting or pricing.

Implications of the guideline for researchers and authors of textbooks:

- Different terms should be used for different concepts. In particular, a distinction should be made between estimated 'most likely effort', 'planned effort' and 'budgeted effort'.
- When conducting surveys or logging estimation information, it must not be assumed that the terminology used is understood, even if it is defined precisely. In-depth studies and triangulation may be needed to ensure that all the data are based on the same understanding of the estimation terminology used.

Implications of the guideline for practitioners:

- Different terms should be used for different concepts. In particular, a distinction should be made between estimated 'most likely effort', 'planned effort' and 'budgeted effort'.
- The estimation of most likely effort should be performed as an independent activity and separated from planning, budgeting and pricing. People in charge of bidding should, for example, not be in charge of the estimation of most likely effort, to ensure that pricing and realism are not mixed. Planning tools should not be used as estimation tools, or, at least, used with great care to avoid a mixing of concerns.

Guideline 2: When assessing estimation accuracy, make sure that the estimated and the actual effort are comparable.

Implications of the guideline for researchers:

- The actual efforts should be adjusted so that they are comparable to the estimated effort with respect to technical and functional parameters before the estimation accuracy is calculated. If functional and quality requirements are not available, the project plan should be investigated and interviews should be used to identify changes in scope and/or quality. If estimates are of types other than most likely effort estimates, they should be transformed to most likely estimates before the accuracy is calculated.
- When estimates cannot be reliably transformed to values that are comparable to the actual result, great care should be taken when using these results, or the projects for which such transformation cannot be performed should be removed from the data set.

Implications of the guideline for practitioners:

- The scope and other assumptions of the estimate of most likely effort should be recorded. The version of the requirement specification, and other documents that the estimate of most likely effort is based on, should be specified.
- Deviation from estimated scope, quality, and development process should be recorded.

The first of the case stories in Section 1 presents an example of how violation of Guideline 1 (estimates of most likely effort not clearly separated from budgeted effort) resulted in a public debate where a research report on overruns of most likely estimates was mistakenly used as evidence of governmental waste of money. An example of the second guideline's importance is presented in the case study described in Section 3. This case study shows how violation of Guideline 2, (no adjustment of estimation accuracy when the estimated and the actual solution differs), would lead to unfair evaluation of estimation ability. More comprehensive estimation terminology guidelines can be found in [2,26,33].

6. Summary

Effort and schedule overruns are serious problems in the software industry. In this paper we argue that the lack of a precise software effort estimation terminology is an important obstacle for the improvement of estimation accuracy. We reviewed the currently most popular software textbooks and a representative set of software estimation research papers and found systematic shortcomings in use of estimation terminology. For example, estimates of most likely effort are frequently mixed with planned effort,

budgets and price. In addition, effort estimation accuracy is frequently measured without adjustments being made for differences in the scope and/or quality assumed when estimating the effort and the system actually implemented.

In order to improve effort estimation accuracy, a more precise terminology for software effort estimation is needed. We provide two simple guidelines for this purpose: (1) Do not mix estimation of most likely effort with planning, budgeting or pricing, and (2) When assessing estimation accuracy, ensure that the estimate and the actual effort are comparable. Although these guidelines are not innovative and might seem obvious, they are nevertheless worth stressing. As this review points out, they are frequently violated.

Acknowledgements

Thanks to Professor Barbara Kitchenham, Professor Ray Welland and the anonymous reviewers for useful comments and suggestions. Also, thanks to the Norwegian Research Council who sponsored this work through the SPIKE and INCO projects.

References

[1] T.K. Abdel-Hamid, Adapting, correcting, and perfecting software estimates: a maintenance metaphor, IEEE Computer 26 (3) (1993) 20–29.
[2] J.S. Armstrong, Standards and Practices for Forecasting, in Principles of Forecasting: A Handbook for Researchers and Practitioners, Kluwer, Boston, 2001.
[3] H. Barki, S. Rivard, J. Talbot, An integrative contingency model of software project risk management, Journal of Management Information Systems 17 (4) (2001) 37–69.
[4] E.J. Barry, T. Mukhopadhyay, S.A. Slaughter, Software project duration and effort: an empirical study, Information Technology and Management 3 (1–2) (2002) 113–136.
[5] F. Bergeron, J.Y. St-Arnaud, Estimation of information systems development efforts: a pilot study, Information and Management 22 (4) (1992) 239–254.
[6] R.H. Berry, R.M. Schoenborn, Estimating requirements for a large, software engineering project (experience with Ada COCOMO on SIDPERS-3), proceeding of the TRI-Ada '92, pp. 375–383, 1992.
[7] B. Boehm, R. Fairley, Software estimation perspectives, IEEE Software 17 (6) (2000) 22–26.
[8] F. Bootsma, How to obtain accurate estimates in a real-time environment using full function points, proc. IEEE Symposium on Application-Specific Systems and Software Engineering Technology, pp. 105–112, 2000.
[9] F. Brooks, The Mythical Man-Month: Essays on Software Engineering, 20th Anniversary Edition, Wesley, 1995.
[10] R. Buehler, D. Griffin, H. MacDonald, The role of motivated reasoning in optimistic time predictions, Personality and Social Psychology Bulletin 23 (3) (1997) 238–247.
[11] P. Coombs, IT Project Estimation—A Practical Guide to the Costing of Software, Cambridge University Press, Cambridge, 2003.
[12] R.A. Cosier, G.L. Rose, Cognitive conflict and goal conflict effects on task performance, Organizational Behaviour and Human Performance 19 (2) (1977) 378–391.
[13] T. DeMarco, Controlling Software Projects, Prentice Hall PTR, Upper Saddle River, 1982.
[14] T. DeMarco, T. Lister, Waltzing With Bears: Managing Risk on Software Projects, Dorset House, New York, 2003.
[15] J.S. Edwards, T.T. Moores, A conflict between the use of estimating and planning tools in the management of information systems, European Journal of Information Systems 3 (2) (1994) 139–147.
[16] R.A. Fleck Jr., Managing programmer resources in a maintenance environment with function points, Industrial Management + Data Systems 98 (2) (1998) 63–70.
[17] A. Gray, S. MacDonnell, M. Shepperd, Factors systematically associated with errors in subjective estimates of software development effort: the stability of expert judgment, Proceedings of the International Software Metrics Symposium, pp. 216–227, 1999.
[18] G.M. Haugnes, Offentlig IT-sprekk for 6 mrd. hvert år, in Aftenposten, j ed. Oslo, 2004.
[19] P. Haynes, B. Henderson-Sellers, Cost estimation of OO projects: empirical observations, practical applications, American Programmer 9 (7) (1996) 35–41.
[20] F.J. Heemstra, R.J. Kusters, Function point analysis: Evaluation of a software cost estimation model, European Journal of Information Systems 1 (4) (1991) 223–237.
[21] K. Heldman, PMP: Project Management Professional Study Guide, SYBEX, Inc., Alamenda, 2002.
[22] J. Hill, L.C. Thomas, D.E. Allen, Experts' estimates of task durations in software development projects, International Journal of Project Management 18 (1) (2000) 13–21.
[23] A.M. Jenkins, J.D. Naumann, J.C. Wetherbe, Empirical investigation of systems development practices and results, Information and Management 7 (2) (1984) 73–82.
[24] J. Johnson, K. Boucher, K. Connors, J. Robinson, The Criteria for Success-Industry Trend or Event, in Software Magazine, vol. February, 2001.
[25] M. Jørgensen, How much does a vacation cost?, Software Engineering Notes 28 (6) (2003) 30.
[26] M. Jørgensen, A review of studies on expert estimation of software development effort, Journal of Systems and Software 70 (1–2) (2004) 37–60.
[27] M. Jørgensen, Realism in assessment of effort estimation uncertainty: It matters how you ask, IEEE Transactions on Software Engineering 30 (4) (2004) 209–217.
[28] M. Jørgensen, D.I.K. Sjøberg, Impact of effort estimates on software project work, Information and Software Technology 43 (15) (2001) 939–948.
[29] M. Jørgensen, K. Moløkken-Østvold, Reasons for software effort estimation error: impact of respondent role, information collection approach, and data analysis method, IEEE Transactions on Software Engineering 30 (12) (2004) 993–1007.
[30] M. Jørgensen, L. Moen, N. Løvstad, Combining quantitative software development cost estimation precision data with qualitative data from project experience reports at Ericsson design center in Norway, Proceedings of Conference on Empirical Assessment in Software Engineering, 2002.
[31] J. Kamatar, W. Hayes, An experience report on the personal software process, IEEE Software 17 (6) (2000) 85–89.
[32] P.G.W. Keen, Information systems and organizational change, Social Impacts of Computing 24 (1) (1981) 24–33.
[33] B. Kitchenham, Software Metrics: Measurement for Software Process Improvement, Blackwell Publishers, 1996.
[34] B. Kitchenham, Procedures for Performing Systematic Reviews, Technical Report, Keele University, Keele, 2004.
[35] B. Kitchenham, S.L. Pfleeger, B. McColl, S. Eagan, An empirical study of maintenance and development estimation accuracy, Journal of Systems and Software 64 (1) (2002) 57–77.
[36] C. Larman, Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and Iterative Development, third ed., Pearson Education, Inc., Upper Saddle River, 2005.

[37] A.L. Lederer, J. Prasad, Causes of inaccurate software development cost estimates, Journal of Systems and Software 31 (2) (1995) 125–134.
[38] M.R. Lind, J.M. Sulek, Undersizing software systems: third versus fourth generation software development, European Journal of Information Systems 7 (4) (1998) 261–268.
[39] S. McConnell, Rapid Development, Microsoft Press, Redmond, 1996.
[40] O. Mizuno, T. Kikuno, K. Inagaki, Y. Takagi, K. Sakamoto, Statistical analysis of deviation of actual cost from estimated cost using actual project data, Information and Software Technology 42 (7) (2000) 465–473.
[41] K. Moløkken, M. Jørgensen, A review of software surveys on software effort estimation, Proceedings of International Symposium on Empirical Software Engineering, 2003 pp. 223–230.
[42] K. Moløkken, M. Jørgensen, S.S. Tanilkan, H. Gallis, A.C. Lien, S.E. Hove, Project Estimation in the Norwegian Software Industry—A Summary, Simula, Technical Report, 2004.
[43] K. Moløkken-Østvold, M. Jørgensen, S. Tanilkan, H. Gallis, A. Lien, S. Hove, A survey on software estimation in the norwegian industry, Proceedings of Metrics '04, 2004 pp. 208–219.
[44] R.S. Pressman, Software Engineering—A Practitioner's Approach, sixth ed., McGraw-Hill, 2005.
[45] J. Ropponen, K. Lyytinen, Can software risk management improve system development: an exploratory study, European Journal of Information Systems 6 (1) (1997) 41–50.
[46] K. Schwaber, M. Beedle, Agile Software Development with Scrum, Prentice Hall, Upper Saddle River, 2002.
[47] M. Shepperd, M. Cartwright, Predicting with sparse data, Proceedings of International Software Metrics Symposium, 2001 pp. 28–39.
[48] Sommerville, Software Engineering, seventh ed., Addison-Wesley, 2004.
[49] J. Sponsky, Joel on Software: and on Diverse and Occasionally Related Matters that will Prove of Interest to Software Developers, Designers, and Managers, and to Those Who, Whether by Good Fortune or Ill Luck, Work with Them in Some Capacity, Apress, Berkeley, 2004.
[50] G.H. Subramanian, S. Breslawski, An empirical analysis of software effort estimate alterations, Journal of Systems and Software 31 (2) (1995) 135–141.
[51] L.M. Taff, J.W. Borchering, J.W.R. Hudgins, Estimeetings: development estimates and a front-end process for a large project, IEEE Transactions on Software Engineering 17 (8) (1991) 839–849.