How Much Does a Vacation Cost? or What is a Software Cost Estimate?

Magne Jørgensen
magnej@simula.no
Simula Research Laboratory, Norway

Abstract: *What is a software cost estimate? Is it the most likely cost, the planned cost, the budget, the price, or, something else? Through comparison with vacation cost estimation and a real-life case we illustrate that it is not meaningful to compare and analyze cost estimates unless it is clear which interpretation is applied. Unfortunately, the software industry, software engineering textbooks and scientific estimation studies do frequently not clarify how they apply the term 'cost estimate'. We argue that this lack of clarity may lead to conflicting estimation goals, communication problems, and, learning problems, and provide recommendations on how to deal with these problems.*

Three families, A, B, C, planned a vacation to Paris. They first decided on their activities and accommodation, and then estimated the costs. All three families estimated that the vacation would cost about $ 3500. From the outside, it may seem that the families had the same expectations about what their vacation would cost. However, there are important differences in what they meant by 'cost estimate':

- *Family A* investigated what most families of their type spend in Paris per week when on vacation. Their cost estimate was therefore meant to be the most likely vacation cost and the family believed that the probability that they would exceed the estimate was about 0.5. Based on their previous experience with 'vacation cost management', and the variance in what other families had spent, they were almost certain that their actual expenditure would fall within ± 25% of their estimate. Family A had produced a 'most likely' or "50-50" estimate.
- *Family B* had no credit cards, limited cash reserves and needed to be risk-minded in their estimation. The family therefore wanted a cost estimate that minimized the likelihood of running out of money. The family believed that the most likely cost was about $ 2500, but added what they believed was a large risk buffer to cover unexpected events. Family B's estimate was meant to be a 'risk-reducing estimate' or a 'budget with low risk of being exceeded'.
- The mother in *Family C* was responsible for the budget and knew that her family was apt to spend a lot of money, should it be made available, e.g., if their estimate was $ 5000, this amount could easily be spent. Her purpose with the cost estimate of $ 3 500 was therefore to reduce expenditure, while still not being perceived as unrealistic by the other family members. Family C's estimate was meant to be a 'cost-reducing estimate'.

The families then took their vacations in Paris and all of them spent about $ 4000.

Question: How good were the families' vacation cost estimates?

Answer: Although the families seem to have made estimates of similar accuracy, there being a deviation of $500 in each case, there are actually huge differences in their estimation performance. Family A underestimated a little, but the deviation was within their expected range. Family B strongly under-estimated the cost and exceeded their budget. In fact, due to poor cost management at the beginning of their vacation they ran out of money, which resulted in their having a disastrous last two days, when they had to borrow money from their country's Embassy to survive. The mother in Family C believed that her cost-reducing goal was met, i.e., that the estimate fulfilled its main purpose.

Now, it may be asked, is vacation cost estimation really relevant to a discussion of software development cost estimation? We believe it is. For example, both types of estimate may be based on vague specifications of the end product, the cost may follow Parkinson's principle (*Work expands so as to fill the time available for its completion),* and the resulting estimation accuracy depends on skilled cost management.

An objection to the relevance of our vacation cost estimation example may be that only Family A provided *estimates*. Family B and C provided *budgets*, i.e., there is a misuse of the term 'estimate' in the example. That is perfectly correct, and in fact constitutes the main point we wish to make in this article.

This misuse and lack of clarity in use of the term 'cost estimate' seems to be typical in the software industry. Based on several industrial studies (see for example [1]), we have observed that the term estimate is applied sometimes to 'most likely software development cost' (ref. Family A), sometimes to 'risk-minded planned

development cost' (ref. Family B), sometimes to 'cost-reducing planned development cost' (ref. Family C) and sometimes even to 'price-to-customer' or 'project bid'. Surprisingly, many of the introductory textbooks and research papers describing cost estimation do not, as far as we can see, clarify which interpretation of 'cost estimate' they use or discuss the problem of differences in interpretation. A typical example is the description of estimation in the popular software engineering text-book written by Pressman [2, p 113] *"... planning involves estimation - your attempt to determine how much money, how much effort, how many resources, and how much time it will take to build a specific software-based system or product."* It is, in our opinion, not obvious from the quotation above, or in the rest of the chapter on estimation, whether *"how much money ... it will take"* refers to the planned cost, the most likely cost or something else. Another example of lack of clarity in the use of the term 'estimation' seems to be present in the frequently quoted Standish Group study, referred to as the Chaos Report (www.standishgroup.com/sample_research/). That study (the 1994-version) reports that "*... the average* (overrun) *across all companies is 189% of the original cost estimate.*" Other parts of the Chaos Report, e.g., the definition of project success as *"... on-time and on-budget, with all features and functions as initially specified,"* suggest that the intended interpretation of 'cost estimate' is a project's 'cost budget'. Unfortunately, it is not obvious that 'cost budget' is the interpretation placed on the term 'cost estimate' by the study respondents. For one thing, the Chaos-questionnaire on the Standish Group's web-site asks for an estimate of the 'project overrun', which may or may not be interpreted as 'budget overrun'. Further, even if we assume that most study respondents interpreted 'project overrun' as 'budget overrun', it is still not clear what *type* of budget (risk-reducing, cost-reducing, etc.) the respondents referred to. In addition, it is not obvious that most *readers* of the report would apply the intended interpretation when, for example, comparing with own estimation accuracy results. Finally, averaging estimation accuracy data over different types of interpretations of 'cost estimate' may not be very meaningful, as illustrated in the vacation cost example. This lack of precise use of essential terms in both the collection and presentation of data reduces, in our opinion, the value of the Chaos Report. (There seem to be other important problems with the Standish Group estimation study, too [3].) In fact, we will argue that the lack of clarity in the use of estimation-related terms probably *contributes* to the chaos described in the Chaos Report. It is possible that the following problems are, to some extent, consequences of the confusion:

- *Estimation problems due to conflicting goals:* Project bids, budgets and most likely cost estimates have different purposes. Ideally, a bid should be low enough to get the job and high enough to maximize profit. The planned or budgeted cost should enable a project to be completed successfully, and motivate efficient work. The estimate of the most likely cost should represent the most realistic use of effort. Several studies report that estimators may have problems separating these goals when making estimates [4-6]. Consequently, if there is a lack of clarity as to what an estimate is meant to be, the actual cost estimate may easily be based on a mixture of goals and interpretations related to price-to-win, budget and most likely cost values. It is well documented that this mixture of goals and interpretations hinders realism [7-9]. This means that the lack of clarity in the use of the term 'estimate' may be an important reason for the typical over-optimism in software development cost estimates.
- *Communication problems*: Without a common interpretation, communication problems can arise. A frequent misunderstanding we have observed is that the estimator means 'most likely cost' and the management (or customer) interprets this as a 'risk-minded planned development cost' or a 'firm price', i.e., believes it unlikely that the cost estimate will be exceeded.
- *Evaluation and learning problems*: As can be seen in the vacation cost estimation example, comparing the estimation accuracy of different types of estimate may be like comparing items from completely different categories, and the consequent lack of clarity will result in difficulties when evaluating estimation accuracy. For example, estimates based on the interpretation 'most likely cost' and 'cost-reducing planned development cost' should be evaluated very differently. While we should expect few cost overruns of 'risk-minded planned development costs', cost overruns of 'most likely cost' should not be unexpected. Not knowing what kind of estimate was intended therefore discourages learning, e.g., it is not possible in cost overrun situations to decide whether the estimate of most likely cost or the risk buffer was too low.

The following real-life case illustrates the communication and evaluation problem related to lack of clarity of interpretation of cost estimate. We analyzed several project experience reports in a large software development organization [10] and found that an important reason for inaccurate cost estimates was, according to the projects' experience reports, incomplete requirement specifications. From the perspective of a project team this opinion is easy to understand. The project may, for example, have had a lot of unexpected work on clarifying what the customer really wanted. On the other hand, when we compared the requirement specification information from the projects' experience reports with the cost estimation accuracy data in the company's experience database we found

indications of the opposite! More often, high estimation accuracy was connected with the *lack* of precise specifications. For example, one project experienced a major change in project scope in the middle of the project period and other unexpected project problems. Nevertheless, the estimation accuracy was very high. The experience report of that project indicated that the amount of delivered functionality and the quality was lower than intended when the cost was estimated, but still within the high-level, vague requirement specification. The high estimation accuracy was, therefore, very much a result of the large flexibility of the requirement specification.

The case documents an apparent inconsistency between the measured cost estimation accuracy, as implemented in the company's experience database, and the projects' own interpretations. While the projects perceived that they had had estimation problems due to instable requirement specifications, the estimation accuracy measures implied that there was no problem. Which viewpoint was correct? Unfortunately, there are reasons to believe there is no real inconsistency in opinions, only a difference in interpretation of cost estimation accuracy. The experience database's measure of cost estimation accuracy was based on the assumption that there was only one category of cost estimates and projects, i.e., the cost estimation accuracy was measured as the difference between actual and estimated cost regardless of type of project and role of cost estimate. The project members themselves, however, did typically not interpret a small deviation between estimated and actual cost in the case of very flexible specifications as indicator of high 'estimation accuracy', but instead as 'good cost control'. For example, assume that a project knows that their original cost estimates was strongly over-optimistic and that the only reason for not experiencing a large cost-overrun was the reduction in functionality and quality enabled by the vagueness of the specification. Then, it is understandable that the project members perceive that there had been a cost estimation problems, in spite of the measured high estimation accuracy.

A consequence of our observation was that we presented, on the meeting where the process improvement activities for the next year were planned, a cost estimation accuracy report where different types of cost estimate and projects were analyzed individually, instead of the usual 'one-accuracy-measure-fits-all'-report. This, we believe, enabled a much more meaningful discussion on where to improve the cost estimation process and other processes.

What should we do to avoid the problems related to lack of clarity in use of cost estimate? We recommend the following actions:

- *Use clearer terms*. For example, replace the vague 'estimated cost' with 'most likely cost' when the meaning is most likely cost. Use 'planned cost', 'price', or 'bid' for the other meanings. As a minimum, ensure that those receiving the estimate understand the way in which you intend the term 'estimate' to be understood.
- *Train people* in the separation of, and differing goals of, most likely cost, planned cost, bid and price.
- *Ensure that the role of a cost estimate is similar before comparing cost estimation accuracy between projects.*
- *Separate the organization's processes* of estimating most likely cost, planning the project, and bidding. Design the process so that realism is the only important goal of the estimators of most likely cost and that they are not affected by planning and bidding issues [6]. For example, assume that a manager in an organization knows how much a customer is willing to pay for a software project. This information should *not* be considered relevant for the estimation of the most likely cost. The estimators should not even possess information about the accepted level of costs. If the most likely cost turns out to be unacceptable in a bidding situation, then the hourly fees or the margin should be reduced. Alternatively, the project should not be started, or should be simplified or re-defined.

References

1. Jørgensen, M. and D.I.K. Sjøberg, *Impact of effort estimates on software project work.* Information and Software Technology, 2001. **43**(15): p. 939-948.
2. Pressman, R.S., *Software engineering: A practitioner's approach*. 2001: McGraw-Hill Higher Education.
3. Moløkken, K. and M. Jørgensen. *A Review of Surveys on Software Effort Estimation*. in *Accepted for presentation on IEEE International Symposium on Empirical Software Engineering (ISESE 2003)*. 2003. Rome, Italy.
4. Edwards, J.S. and T.T. Moores, *A conflict between the use of estimating and planning tools in the management of information systems.* European Journal of Information Systems, 1994. **3**(2): p. 139-147.

5. Goodwin, P., *Enhancing judgmental sales forecasting: The role of laboratory research*, in *Forecasting with judgment*, G. Wright and P. Goodwin, Editors. 1998, John Wiley & Sons: New York. p. 91-112.
6. Jørgensen, M., *A Review of Studies on Expert Estimation of Software Development Effort.* To appear in: Journal of Systems and Software, 2004.
7. Cosier, R.A. and G.L. Rose, *Cognitive conflict and goal conflict effects on task performance.* Organizational Behaviour and Human Performance, 1977. **19**(2): p. 378-391.
8. Keen, P.G.W., *Information systems and organizational change.* Social Impacts of Computing, 1981. **24**(1): p. 24-33.
9. Buehler, R., D. Griffin, and H. MacDonald, *The role of motivated reasoning in optimistic time predictions.* Personality and social psychology bulletin, 1997. **23**(3): p. 238-247.
10. Jørgensen, M., L. Moen, and 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 the conference on empirical assessment in software engineering*. 2002. Keele, England: Keele University.