Communication of software cost estimates

Magne Jørgensen
Simula Research Laboratory &
University of Oslo
Lysaker, Norway
+47 924 333 55
magnej@simula.no

ABSTRACT

The meaning of an effort or cost estimate should be understood and communicated consistently and clearly to avoid planning and budgeting mistakes. Results from two studies, one of 42 software companies and one of 423 individual software developers, suggest that this is far from being the case. In both studies we found a large variety in what was meant by an effort estimate and that the meaning was frequently not communicated. To improve the planning and budgeting of software projects we recommend that the meaning of effort estimates is understood and communicated using a probability-based terminology.

Categories and Subject Descriptors

D.2.9 Cost estimation, K.6.1 Project and people management, K.6.3 Software management

General Terms

Management, Measurement, Experimentation

Keywords

Cost estimation, terminology, communication of estimates

1. INTRODUCTION

When a software project has not yet been started, there are several possible effort usage outcomes. The possible use of work-effort is for this reason not a single value, but rather a distribution of possible values where some values are more likely than others. An example of a possible distribution of software project effort outcomes with connected probabilities is displayed in Figure 1.

As can be seen from Figure 1, different types of effort estimates differ in interpretation and probability. The most optimistic use of effort, which in the example has a very low probability, is 70-90 man-months. The most likely effort (the highest probability or the mode of the distribution) is 90-110 man-months. The estimate that assumes a 50% probability not to be exceeded (the median or p50-estimate) is 110-130 man-months. To get an estimate that is 85% likely not to be exceeded (a p85-estimate), with may be useful for budgeting or pricing purposes, the effort should be in the interval 130-150 man-months. Not knowing or communicating which of the above meanings that is meant by an effort estimate can create problems. Unfortunately, the results of the studies in this paper suggest a great variety in use and a poor communication of the term effort estimate in software project estimation contexts.

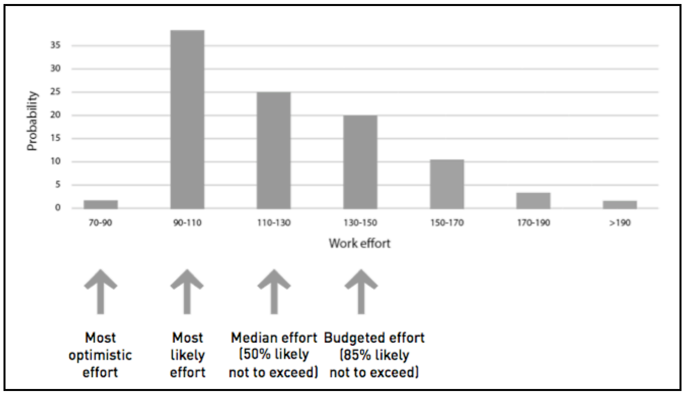

Figure 1: Example distribution of effort usage and types of estimates (work-effort in man-months)

As can also be seen from the example in Figure 1, the distribution of effort is not symmetric around the most likely effort. This reflects a situation where it is more likely to use more than it is to use less effort than the most likely use of effort and where the difference between the most likely (90-110 man-months) and the most optimistic (70-80 man-months) effort is much less than the difference between the most pessimistic (>190 man-months) and the most likely effort. This asymmetry is present in many software project contexts, as illustrated by Figure 2. Figure 2 displays the actual efforts in percent of the estimated most likely effort, i.e., 100% x (Actual effort/Estimated effort), of 42 software projects completed by Norwegian software companies [1] (data set available upon request to the author). As can be seen, the distribution is skewed and has a long tail towards effort overruns. In particularly important, the use of "most likely effort" or the "median" (p50) estimate as the planned effort would on average lead to effort overruns. Similar findings are reported, amongst others, in [2].

In a previous paper on the use of estimation terminology, see [3], we observed that key software engineering textbooks and frameworks did not contribute much to a precise communication of what is meant by an effort estimate. Even frameworks that only focus on cost estimation are not always precise about what is meant by an estimate. AACE (Association for the Advancement of Cost Engineering), for example, has a document on cost engineering terminology that states that a cost estimate is "*A compilation of all the probable costs of the elements of a project or effort included within an agreed upon scope.*" (http://www.aacei.org/non/rps/10s-90.pdf) As we see it, this definition does not say much about whether a cost estimate should

Conference'10, Month 1–2, 2010, City, State, Country.

be used to denote the "most likely", the "p50" or some other probable effort values.

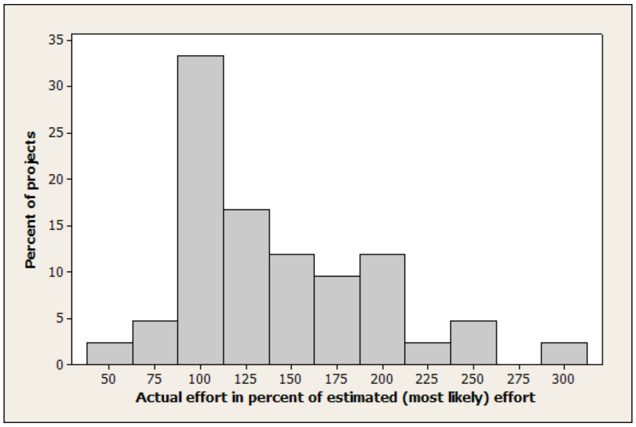

Figure 2: Actual effort in % of estimated most likely effort

A lack of precise communication in the software industry has been experienced, but only informally reported, in [4]. We hypothesize that this lack of clear terminology in the software engineering literature reflects a substantial and unfortunate lack of clarity among software professionals.

Notice that there are several software engineering books and articles that use a precise estimation terminology, e.g. [5-7]. In addition, the effort estimates produced by formal estimation models derived from analysis of actual effort of completed project do to some extent have a precise meaning. Models derived from traditional linear regression analysis, for example, may be said to provide the mean estimate conditional on the values of the independent variables. This would for a symmetric, but hardly for a skewed, distribution be close to a p50-estimate. With log-transformation of variables, which is common in effort estimation model constructions, the interpretation of the model-based estimates becomes more complex (e.g., geometric rather than arithmetic mean), but may also be close to a p50-estimate interpretation. Analogy-based estimation models may produce estimates with interpretations close to p50-estimates when the median or mean of the identified analogies are used to calculate the estimated effort. When formal effort estimation models take expert judgment as input, e.g., uses judgment-based productivity values, the interpretation of the produced effort estimate depends on the meaning of this input, e.g., whether it is the "ideal" or "median" productivity, and we may have many of the same interpretation problems as those reported in this article.

In this paper we focus on judgment-based effort estimates, which is by far the type of estimates most used in the software industry [8]. To test our hypothesis of inconsistency and lack of clarity in communication of effort estimates we examined how the term "effort estimate" was used by 42 software companies (Section 2) and by 423 individual software developers (Section 3). We then conclude (Section 4).

2. A STUDY OF SOFTWARE COMPANIES

2.1 Study Design

We invited 42 software companies from Eastern Europe (mainly Russia, Ukraine and Romania) and Asia (mainly India) to participate in the study. All companies estimated the development effort of the same two software projects (requirement specifications will be made available upon request). One of the projects was a relatively simple web-application and the other was a much more complex inventory management system. The companies were paid ordinary fees for their estimation work and told that they were providing second opinion effort estimates for the purpose of comparing their effort estimates with that of other companies. The companies spent on average about one workday for each project estimate. To avoid that the context would be interpreted as a competitive bidding situation, we informed them that they would not be asked to complete the projects and we were only interested in high quality estimation work. We asked them to estimate how much effort they would need to complete the projects, but gave no precise instruction about how to interpret what this meant in terms of type of estimate. As part of the estimation work we asked the companies to document essential assumptions of their estimate, describe the estimation process and assess the uncertainty of the effort estimate. The uncertainty of the estimate of the total effort was assessed through a request to assess how probable that the actual total effort would be:

I) Less than 50% of the estimated effort

II) Between 50 and 75% of the estimated effort

III) Between 75 and 100% of the estimated effort

IV) Between 100 and 125% of the estimated effort

V) Between 125 and 150% of the estimated effort

VI) Between 150 and 200% of the estimated effort

VII) More than 200% of the estimated effort.

Of particular relevance to our analysis is how probable the companies thought it would be that the actual effort exceeded the values they had termed their effort estimates. This probability, which we term *ProbExceedEst*, is defined as the sum of the probabilities for the intervals IV, V, VI and VIII. We categorized an effort estimate with *ProbExceedEst* higher than 80% as "ideal effort" (highly likely to use more effort than expected), between 61 and 80% as "most likely effort" (assuming a skewed distribution where most likely effort is likely to be exceeded), between 41 and 60% as "median effort" (represent a probability of exceeding of about 50%), and 41% or lower as "risk averse effort" (unlikely to be exceeded). As can be seen, a very accurate estimate (100% of estimated effort) may be put in either category III) or IV). The motivation behind including 100% in both categories was to make the companies decide on which side of the estimate they assumed that the effort of even quite accurately estimated projects (+/- 25% of estimated effort) would be. Our hypotheses were that: i) The companies would differ much in their interpretation of "effort estimate", as measured by *ProbExceedEst*, and ii) The companies would typically not communicate clearly what they meant by "effort estimate".

2.2 Results

Table 1 shows the distributions in interpretation of effort estimate among the companies, as measured by the probability of exceeding the value they had provided as their effort estimate (*ProbExceedEst*).

Table 1: Interpretation of estimate, as indicated by probability of exceeding the estimate (*ProbExceedEst)*

ProbExceedEst	**Complex system**	**Simple system**
Ideal effort (81-100%)	36%	28%
Most likely effort (61-80%)	23%	27%
Median effort (41-60%)	28%	24%
Risk averse effort (0-40%)	13%	21%

Table 1 reports a substantial variance in how the companies assessed the probability of exceeding their estimates. This data, consequently, gives a first indication of a substantial difference in what the companies meant by their effort estimate. Interestingly, the companies typically used a more optimistic interpretation of effort estimate, i.e., they think it is more likely to exceed the estimated effort, when estimating the complex system. This suggests that the meaning, as measured by *ProbExceedEst*, of an effort estimate did not only vary between companies, but also within the same company, dependent on the complexity of the project. As we will see in Section 3, an independent study, supports that the distribution of interpretations, as denoted by "ideal effort", "most likely effort" etc. in Table 1 reflects the variance in interpretations among software professionals.

We added an analysis where we included only the categories V, VI and VII, i.e., we included a measure of the assessed probability of exceeding the estimate with more than 25%. As can be seen in Table 2, there is a substantial variation between the companies for this measure, too. Although this variation may be affected by a difference in estimation skill, i.e., less skill leads to higher level of uncertainty, we believe it also reflects a difference in what is meant by the estimate. When, for example, 23% of the companies believes that it is more than 40% likely to overrun the estimate of the complex system with more than 25%, this suggest that the effort estimate is quite idealistic.

Table 2: Probability of exceeding the estimate with more than 25%

ProbExceedEst	**Complex system**	**Simple system**
41-100%	23%	8%
21-40%	21%	18%
11-20%	23%	19%
0-10%	33%	55%

We examined the companies' description of their estimates and estimation processes as provided in their estimation documentation. Only six out of the 42 companies included a description of the effort estimate or the estimation process that enabled us to interpret the meaning of "effort estimate" in terms of the categories in Table 1. Among those six companies, two claimed to have used the interpretation "most likely effort", and four the interpretation "median effort" (p50). In addition, one company claimed to have used the interpretation "mean" effort (expected value) for their effort estimates. The claimed interpretation for all seven companies was consistent with the interpretation as indicated by their *ProbExceedEst*-value.

Our data provide support for a substantial difference in interpretation of the term "effort estimate", as measured by *ProbExceedEst*, and a frequent lack of communication of what is meant by this an effort estimate. Clearly, the results from a study of only 42 outsourcing companies can hardly be used to make strong claims about the general state of practice in the software industry.

Further support for our findings is provided by our own experience from working with software companies or analyzing estimation work of completed software projects. We have experienced that it is typically hard to know what is meant by the provided effort estimates.

3. A STUDY OF SOFTWARE DEVELOPERS

3.1 Study Design

In the first study we derived the meaning of the effort estimate from how likely the companies thought it was to exceed their effort estimates. To check the interpretations and communications of effort estimates more directly, we designed a study where we instructed the software professionals to be explicit about what they meant by their estimates. We invited fourteen software companies (only minor overlap with the companies in the first study), from Romania, Ukraine and Poland, to participate with their software developers. As in the first study, the software developers were paid ordinary fees for their estimation work. In total 423 software developers participated. All developers had at least one year of experience and had experience in the estimation own and/or other developers' software development work. The inclusion of, up to 166, software professionals from the same companies enabled us to analyze whether there would be the same interpretation of effort estimate within the same company. Our hypothesis, based on the results from the first study, was that this would not be the case.

Following the effort estimation of four software development tasks, we gave the developers the following instruction:

You have just estimated the number of work-hours you think you need to develop and test four different software systems. Please select the description below that you think is closest to what you have meant by an effort estimate in the previous four estimation tasks.

a) *Effort estimate = Number of work-hours I will use given that I experience almost no problems.*

b) *Effort estimate = Number of work-hours I will use given that I experience no major problems.*

c) *Effort estimate = Number of work-hours I most likely will use.*

d) *Effort estimate = Number of work-hours where it is about just as likely that I will use as it is that I will use less effort than estimated.*

e) *Effort estimate = Number of work-hours where it is unlikely that I will use more effort than estimated.*

f) *Effort estimate = Number of work-hours based on my expert judgment/feeling of how many work-hours I will use. I find it difficult to decide about the exact meaning of the estimate.*

g) *None of the above descriptions is close to what I typically mean by an effort estimate. (Please specify briefly what you typically mean by an estimate in the box below.)*

In the following, we term categories a)+b) as "ideal effort", c) as "most likely effort", d) as "median effort", e) as "risk averse effort", f)+g) as "unknown/other". These categories are meant to correspond with the categories we introduced in the first study.

3.2 Results

The responses are summarized in in Table 3, which also include the responses from the first study. To ease the comparison we combine (take the average of the responses for the simple and the complex system) and adjust (multiply the original values with 78%=100%-22%, due to missing category of "unknown/other" in the first study) the percentages from the first study.

Table 3: Distribution of meaning of effort estimate

Meaning of effort estimate	**Software developers (second study)**	**Software companies (first study)**
Ideal effort	37%	25%
Most likely effort	27%	20%
Median effort	5%	20%
Risk averse effort	9%	13%
Unknown/other	22%	Category not used

As can be seen, there are both similarities and differences between the distributions from the first and the second study. Both studies observe a substantial variation in the meaning of the values described as effort estimates. Both studies have as the most frequent understanding of effort estimate the "ideal effort", where "ideal effort" is understood as the assumption of "almost no problems" or "no major problems" in the second and as a probability of exceeding the estimate of more than 80% in the first study. Very few (only 5%) selected the median effort (p50) as the interpretation of the estimated effort in the second study, while this seemed to be a quite common interpretation in the first study. This could be a result of the difference in contexts of the two studies. The first study had a more traditional project team estimation context, while the second study asked for the estimation of own effort completing smaller projects/tasks. Rationally speaking, a difference between project team work and own work should not give a difference in interpretation of effort estimates. Empirical evidence suggest however that people tend to think more idealistically when estimating own work [8].

It may be argues that many of the developers did not really know what they meant by their estimates and just gave a best guess of what they might have meant. If this is the case, the problem may be even larger than problematic communication of estimates. Clearly, a necessary condition for precise communication is that the person producing the estimate knows what he/she has meant by it.

We had hypothesized that there would be a large variance of estimation interpretations even within the same company. Table 4, which displays the estimation interpretations for the developers within the five companies with at least 20 software developers participating, supports this hypothesis.

Table 4: Distribution of meaning of effort estimate within the same company (n = number of participants of a company)

Company	**A** **n=20**	**B** **n=79**	**C** **n=32**	**D** **n=166**	**E** **n=21**
Ideal effort	50%	34%	38%	35%	24%
Most likely effort	30%	29%	13%	31%	33%
Median effort	10%	4%	6%	6%	5%
Risk averse effort	0%	8%	6%	8%	19%
Unknown/ other	10%	25%	37%	20%	19%

As can be seen, there is a large variation of the meaning of the term effort estimate even within the same company. For further evidence, we gave the same questionnaire about the meaning of an effort estimate to two Norwegian software companies, as part of company internal estimation seminars. The distributions of effort estimation interpretations of these two companies were very similar to those in this study.

A variation of interpretations of effort estimates may not be a large problem if those producing them communicate what they mean by it. In the first study we found that this was seldom the case. We examined this issue in this study as well, by asking the software developers:

When you give your estimate to someone else (for example your project manager), do you inform about what you mean by your estimate?

A summary of the responses gave that 43% of the developers claimed to always, 36% sometimes 17% seldom and 3% never informed about what they meant by an effort estimate. One percent did not respond to the question. We should be careful about interpreting such responses since the results may easily be biased towards responses providing what they think is the normatively correct answer. It is nevertheless reasonable safe to conclude that the responses suggest that there were no established practice of explaining what is meant by an estimate that ensured the removal of the potential problems of different interpretations of effort estimates.

4. CONCLUSION

A mature engineering discipline should use concepts that are clearly defined, well understood, and consistently used and communicated. The results from the two studies reported in this paper suggest that this is not always the case for the essential term effort estimate. A variation in and poor communication of what is meant by effort estimate may easily contribute to effort and cost inaccuracies and cause planning and budgeting problems for software projects. Cost overrun may for example occur when the intended meaning is the ideal or the most likely use of effort, while the receiver of the estimate, e.g., the project manager or the client, understands the estimate as something much less likely to be exceeded, e.g., a p50 estimate. It is then not necessarily an inaccurate estimate, but rather a poor communication of the estimate that causes the overruns of plans and budgets. In addition, when evaluating the estimation accuracy and combining several estimates into one total estimate, it is essential to ensure that the meaning of the estimates are well understood and consistently used. Otherwise the total estimate will be based on “adding apples and oranges”.

To improve the situation we think it is essential that:

- Estimate providers should document and communicate clearly what is meant by an effort estimate.
- The software companies and their clients should be trained in the use of a probabilistic terminology for effort estimates (see Figure 1 for an example and [9] for an approach for this).
- Estimate users should ensure that they understand what is meant by an estimate, including the assumptions made and the type of estimate provided.

5. REFERENCES

[1] K. Moløkken-Østvold, M. Jørgensen, S. S. Tanilkan, H. Gallis, A. C. Lien, and S. E. Hove, "A survey on software estimation in the Norwegian industry," in *10th International Symposium on Software Metrics*, Chicago, IL, 2004, pp. 208-219.

[2] L. Todd, "Schedule estimation and uncertainty surrounding the cone of uncertainty," *IEEE Software,* vol. 23, no. 3, pp. 48-54, 2006.

[3] S. Grimstad, M. Jorgensen, and K. Molokken-Ostvold, "Software effort estimation terminology: The tower of Babel," *Information and Software Technology,* vol. 48, no. 4, pp. 302-310, 2006.

[4] D. Basten and M. W, "A Current Assessment of Software Development Effort Estimation," presented at the Empirical Software Engineering and Measurement (ESEM), 2011.

[5] B. Kitchenham and S. Linkman, "Estimates, uncertainty, and risk," *IEEE Software,* vol. 14, no. 3, pp. 69-74, May-June 1997 1997.

[6] T. DeMarco and T. R. Lister, *Waltzing with bears*: Dorset House Pub., 2004.

[7] S. McConnel, *Software Estimation: Demystifying the Black Art (Best Practices*: Microsoft Press, 2006.

[8] T. Halkjelsvik and M. Jørgensen, "From origami to software development: A review of studies on judgment-based predictions of performance time," *Psychological Bulletin,* vol. 138, no. 2, pp. 238-271, 2012.

[9] M. Jørgensen, "Realism in assessment of effort estimation uncertainty: It matters how you ask," *IEEE Transactions on Software Engineering,* vol. 30, no. 4, pp. 209-217, 2004.