A metric of estimation quality

by TOM DEMARCO
Yourdon Inc.
New York, New York

ABSTRACT

This paper reports on a metric of the estimating process called Estimating Quality Factor (EQF). EQF is used to gauge progress in estimating technique and to provide an easily measurable standard for good estimating.
INTRODUCTION

You cannot control what you cannot measure. Anything you do not measure is almost certainly out of control. To say that the estimating process for software is out of control would be relatively kind: In a recent metric survey\(^1\) I conducted, estimated development efforts were plotted against actual efforts, giving the dismal result shown in Figure 1. The caption of Figure 1 may be overly optimistic, because a number of participants dropped out of the survey before completion, and there is every reason to suspect that those who drop out of a productivity survey tend to be among the lower-than-average performers.

Software cost estimates are notoriously poor. In a recent chemical plant construction project, support software was delivered at a cost nearly 400% of the original estimate; the construction effort was delivered within 9% of its estimate. One very evident difference between the methods used by construction estimators and those used by software estimators is that construction estimators keep rigorous and formal records on the quality of their estimates; software estimators do not. A prevailing software industry standard is to lose estimates the moment they are found to be wrong so that, by the end of a project, no one can say for sure just how bad the original estimates were.

As the software discipline matures, we will need to adopt some of the methods used by estimators in other fields in order to be able to approach their performance. In particular, we will need to formulate a precise and verifiable measure of estimation quality and use that measure to analyze and improve the estimating process. This paper describes one such measure.

ASSESSING THE QUALITY OF AN ESTIMATE

The quality of an estimate must be judged with respect to the duration of the unknown. Estimates generated when management typically requires them—first microsecond of the project—should be expected to have wide tolerances. Later estimates, say from the project 50% point, should have considerably narrower tolerances. Any assessment of estimation quality must, therefore, take into account the point in time at which the estimate was produced.

**First Rule:** Estimate quality increases monotonically with **accuracy** (how closely the estimate predicts the actual) and with **earliness** (how soon in the project the estimate is produced).

The first, and in many ways most crucial, estimates are commissioned at the very beginning of a project. There are so many unknowns at this point that the tolerance of estimates is unacceptably high. Most projects are begun with estimates that (time will prove) have a tolerance of \(-20\%\, +300\%\). Because tolerances are often not expressly stated, managers assume an expected deviation of \(-0\%\, +0\%). They deal with the problem of unacceptably wide tolerances by ignoring it. A manager who takes explicit account of the wide tolerances of early estimates, however, is obliged to narrow the risk as the project goes on. This means that he/she must begin to consider estimating as a **continuing process**. A project may begin with the understanding that cost may exceed benefit by a factor of 300%, but it cannot be continued on that basis. Subsequent estimates will have to narrow the unknown so that go/no-go decisions can be made intelligently.

**Second Rule:** Estimate quality must be assessed not on the initial estimate alone but on the dynamic characteristic of successive estimates and their convergence to actual results.

This means that key project parameters will have to be estimated and re-estimated throughout the project and a formal history maintained. The history might take this form:
ABX Project: Estimated Debugging Effort (Man-Hours)

<table>
<thead>
<tr>
<th>Date of Estimate</th>
<th>Estimated Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/15/80</td>
<td>2,000 hrs</td>
</tr>
<tr>
<td>3/15/80</td>
<td>3,500 hrs</td>
</tr>
<tr>
<td>6/15/80</td>
<td>5,500 hrs</td>
</tr>
<tr>
<td>8/15/80</td>
<td>5,000 hrs</td>
</tr>
<tr>
<td>9/15/80</td>
<td>4,750 hrs (actual)</td>
</tr>
</tbody>
</table>

ESTIMATING QUALITY DEFINED

The metric *Estimating Quality Factor* (EQF) is defined by applying the two rules to the estimating history in the following fashion:

\[
EQF = \text{Reciprocal of the average absolute deviation}
\]

where the deviation between estimate and actual is expressed as a percentage of the actual. This formulation is more easily understood from a graphic representation of the estimating history, as shown in Figure 2. EQF is now seen to be the area under the actual result \((A \times T)\) divided by the shaded area. High EQF numbers imply good estimates. The starting point for computation of EQF is taken arbitrarily as the point at which 10% of elapsed time has been used up. Obviously, EQF can be assessed only after the estimated parameter becomes known.

EARLY EXPERIENCE WITH THE EQF METRIC

For 17 projects studied in Reference 1, average EQF was slightly below 4. The estimating technique used in these projects seems to have been the traditional gut-feel approach. I concluded that an EQF of 4 or lower characterizes the current state of the art of software estimating, and it is a reasonable standard against which to test new estimating techniques. Techniques that raise EQF substantially above 4.0 can be considered improvements.

Experiments in six client companies involved the establishment of politically separated groups of estimating experts according to the method described in Reference 2. These estimating teams did all estimates within their organizations. Project personnel did no estimating and estimators did no development. Estimators were instructed to track EQF for 30 key project parameters and to endeavor to maximize average EQF. They had no incentives beyond this. Explicitly stated, they had no incentive to come up with optimistic estimates, because such estimates would not reflect on their own development skills; no incentives to come up with padded estimates, because estimates that are above the actual reduce EQF just as much as those below; and no disincentive to frequent re-estimation. The estimators applied metric techniques taken from References 2, 3, and 4 as well as from other sources. Average EQF for the experiment improved over the 18 months during which data were collected, from an initial value of 3.8 to 10 or higher.

CONCLUSION

Estimating quality can only improve if we adopt some way to measure and track it. The measure selected should depend on the degree of convergence of estimates to actuals over time. The EQF metric is a palatable metric of estimating quality, at least for some organizations.* Improvement of average EQF in the experiment is the only evidence I know that anyone has ever made progress in learning to estimate software development.

REFERENCES


*One of the participating companies insisted on redefining EQF to weight earlier estimates somewhat higher. Although their formulation was slightly different, it was also a measure of convergence of estimate to actual over time. The specific formula seems to me to be not so important as the idea that there be a predefined formula for estimating quality.