Familiar Metric Management -
Negotiate from a Fact-Based Position

Lawrence H. Putnam

Ware Myers

"Phil, did you see Steve McConnell's article "How to Defend an Unpopular Schedule" in this IEEE Software magazine?" John asked, waving the May issue above his head. "He has a lot of good ideas."

John is a software engineer and Phil is his project leader.

"I saw it," Phil replied. "Haven't read it yet. I suppose he is in favor of adequate schedules."

"Of course he is," John said. "He is a software engineer himself, writing in a software magazine."

"How does he figure out what an adequate schedule is?" Phil asked.

"Well, he doesn't really go into that," John responded. "He seems to assume that software people will know how long a project will take and that marketing types and customers will want it sooner."

"Right—sooner, better, cheaper," Phil said. "So what else is new?"

"He has a lot of ideas on how software people can negotiate better schedules," John said. "For instance, he says:

'Batten down the hatches and endure the thunderstorm of an unwelcome estimate early in the project instead of the hurricane of schedule slips and cost overruns later on.'"

"You'd probably get washed overboard," Phil retorted. "Of course, you might wash up on a tropical island populated by a tribe of Amazons."

"Not any more," John retorted. "I think Amazons were sent to

law school."

Phil and John are our resident jokesters. Nevertheless, McConnell manages to write three pages on negotiation tactics without once using the word metrics. He does use the term, objective criteria, as in: "You search for objective criteria you can use to break the deadlock."

A paragraph further on he says: "One key to keeping schedule negotiations focused on principles and not just desires is to insist on a rational estimation process."

It seems to us that, before negotiations even begin, software people should focus on three tenets:

(1) Base the estimating procedure on accepted metrics;

(2) Use the metrics in an established procedure to set the estimate;

(3) Distinguish between the computed estimate and the final bid.

Base estimate on facts. Many organization still lack agreement within their own ranks as to the facts pertinent to estimating software development. It is no wonder in this situation that marketing people and executives feel free to impose their own wishes on the proposed schedule and cost. If no one has any accepted facts (derived from metrics), then the older people in the management structure at least have the benefit of more years of experience than the generally younger software developers. Moreover, they have the sheer power that comes from higher positions in the hierarchy.

Getting adequate facts, unfortunately, is not a simple matter. We can simplify what has to be done to six steps— perhaps oversimplifying the complexity of the metric process:

(1) Select from the great number of metrics that have been employed the minimum set necessary for estimating. As you know, from past columns we believe that estimating can be done quite nicely from just four common metrics: size, effort, schedule time, and defect rate.

(2) Define each metric for your own organization. If you can use industry-wide definitions, you can compare your operations to other's experience.

(3) Discipline your organization to collect these metrics.

(4) Left to their own devices, people may twist the collection of metrics to avoid personal responsibility. State clearly that metrics are for planning purposes, not merit review purposes. Keep the two functions strictly separate.

(5) Store the metrics on past projects in a form, such as a database, readily accessible to those who need and use the data. Only heroes go out in the back warehouse to dig through old paper-based files. (The last hero departed for Microhard (!) about 1990.)

(6) Make sure everyone in your organization—marketing, managers, executives, as well as software people— appreciate the impartial nature of this data.

(7) Put out white papers or other forms of communication to customers, in-house or external. They, too, need to have confidence in your facts.

Estimate with a method. Estimating is the process of figuring out how long and how much effort it will take to do something. If you have done something very similar before, then the data on your previous projects, if you have this data on hand, gives you a pretty good idea of how long and how much effort the proposed project will take.

If the new project is different, but in the same application area, you can still make an estimate. The estimate will be less precise.

If you have never done anything like it before, the new project is really research. You have little basis for an estimate of how long or how much effort it will take.

Of course, there is a continuum—from projects your organization could do in its sleep to completely unique work. You need a method of characterizing where along this continuum a project lies.

There are, basically, four factors in a software macro-estimating method: size of the project, productivity of the process, schedule time, and effort. When you sit down to estimate, you don't know size, time, or effort. You may know process productivity if you have kept records of past projects. In the case of completed projects, you know size, time, and effort. From this data, you can calculate process productivity.

There is no formulaic way to estimate size beforehand. Your best and most experienced developers have to spend some time considering the requirements, specifications, and functional design, depending on how far along the project is.

Given an estimate of size and a calibrated value of process productivity (and two equations), you can calculate time and effort. To the extent that the values of size or process productivity are uncertain, you can enter them into the computation in the form of a range. The output values for time and effort then appear as a range. That range can be converted by statistical methods to a risk probability. For instance, there is a 75-percent probability of completing this project within x months and y person-months.

The foregoing is a brief outline of the estimating method that we advocate. However, any method based on facts and a known procedure is better than unstandardized guesswork based upon few facts. Together, facts and method give customers, marketing people, and executives confidence in the merit of estimates. They minimize the range over which representatives have to conduct negotiations.

Distinguish estimates/bids. Perhaps somewhat arbitrarily we consider an estimate to be something arrived at through a formal process based upon metrics. The result is not a point number. It may be a range—quite narrow if the proposed work is well known, but quite wide if the work is unfamiliar. Or it may be a point number modified by a risk value, as noted above.

At the current state of the development of civilization, as it happens, many customers want a single number for cost and another single number for schedule. At least, they say they do, and it is quite customary to give them single numbers. We call these numbers the bid.

Going from the estimate to the bid is the province of management, marketing, and the customer. For example, management may choose to "buy" a job by deliberately bidding low, perhaps to break into a new field. It should expect to lose money on the job, not hope for a superhuman effort on the part of the developers to pull this chestnut out of the fire.

We have noted, however, that many customers, while accepting point bids, do not really expect them to come true. Some customers hold back some of the funds

available with the conscious plan of making them available on a follow-on contract to complete the project. They fail to get excited when the project runs a few months longer than bid.

In house, projects are often funded on a level-of-effort basis, and they take as long as they take. Few regard the planning numbers for time and effort as graven on stone.

These are really dodges, based on management's experience that many projects do run over time and over budget. Everyone would be much happier if projects could be estimated and bid with reasonable precision.

In fact, they can be. We have now accumulated a database of about 4000 completed projects from companies in Europe, United States, Canada, and eastern Asia. In more than 600 cases the data includes both the preliminary estimates and the final schedule and effort data. About one-third were completed within these targets; another third were completed within 125 percent of target. Only the final one third exceeded 125 percent of target and might be labeled as out of control.

It is a fundamental property of human character to be able to believe in all earnestness, after having missed twenty-two consecutive monthly schedules, that there is no reason whatsoever to question that the next month's schedule can be met.
　　　Norman Augustine in his book, *Augustine's Laws*, Penguin Books, 1987.