Counting Boulders and Measuring Mountains

Accuracy and the measurement of peaks and projects.

There is an apocryphal story about the first expedition to attempt to accurately measure the height of the mountain in the Nepalese Himalayas called Chomolangma (“Mother of the Universe”) by the Tibetans, Sagarmatha (“Goddess of the Sky”) by the Nepalese, and Peak XV (“Peak 15”) by the somewhat more phlegmatic British. It was in the year 1856 when an expedition, headed by the then Surveyor-General of India, trekked into the mountains of northern India and up into Nepal. The survey team did not climb to the summit of the target mountain, though they did get to a creditable altitude, higher and closer to the mountain than earlier measurements. Employing their surveying craft they took a number of measurements around the mountain, performed their calculations and, to their horror (according to the tale), determined the height of the mountain to be exactly 29,000 feet.

DIVIDE BY 10

We humans make an interesting assumption about measurements. If they happen to fall on a number divisible by the number 10; or better—any large power of the number 10—we assume certain things about that number, like its precision and its importance. Anything divisible by 100 is always very interesting. In the U.K., if you are fortunate to live to 100 years old, the monarch will send you a birthday telegram. For some reason, the Queen will most likely ignore both you and all your birthdays up until you reach that magic number. Larger powers of 10 may attract even more significant attention, the millennium celebrations several years ago being a good example. Interestingly, while most of us seem to accord both great significance and a certain inaccuracy to any multiple of our primary base number in any unit, the same also seems to attract its complement of precision pedants. How many letters in newspapers did you see complaining that the millennium did not actually start until the end of 2000? It was the inferred imprecision of the round number that so horrified the
The Business of Software

surveyors of Peak XV.

Software projects share some similarities with mountains in a metaphorical sense, but we usually go about measuring software projects in a different, and sometimes not-quite-logical way.

**Total Tasks and Evaluate Effort**

A project manager (PM) estimating a project under consideration will typically do the following:

- Obtain the most detailed description of the project that can be defined in the time available. This will usually vary from relatively incomplete to woefully inadequate.
- Collect as many experts as are available and are willing to spend the time “scoping” the project. This is not usually a popular task.
- Identify, sometimes in excruciating detail, all the tasks thought necessary to complete the project.
- Arrange these presumed tasks in a presumed order, identifying any presumed dependencies. This order is usually based on the PM’s experience, which is based on a set of previous projects that are, of course, somewhat unlike this project.
- Using the aforesaid experts and managerial experience, try to establish how “big” each task is. This may be done in a variety of ways, the most common involving an educated guess.
- Summing the tasks then gives the project’s total effort. Calculating the duration along the expected critical path of most-dependent tasks gives the project duration, and adding up all the people necessary to complete the tasks at any given point in time gives the staffing profile.

This is all well and good.

The task-based work breakdown structure (WBS) approach to estimating a project can be a solid management practice—under certain circumstances. Problems arise when we don’t know what the tasks are because this project is new. While it may share some similarities with prior projects, at least part of the system being built by this project must always be “new.” We could argue that the “old” part of the project—those tasks that are very similar to prior projects—really shouldn’t take too much time or require a great deal of effort; we have already done them at least once after all. If we really were able to learn from our mistakes on previous projects, the only difficult part of this project would be the stuff we had never done before. The problem is that it is the very same new stuff that is the most difficult thing to estimate.

The level at which a WBS is created often gives the appearance of precision, because it is very detailed. But it may not be precise.

**How Project Managers Think**

The reason PMs reach to a task-based WBS approach when estimating projects is simple: it is how they view a project. Ask a PM to describe or even think about a project, and you’ll get a list of tasks. To a PM, a project is a set of tasks. This viewpoint is not necessarily shared by other stakeholders. Most customers know little and care less about the tasks necessary to build a system. To a customer, a project is the user interface, the functionality, and the price and schedule for delivery. To a developer a project is more akin to a complex technical puzzle. To an executive a project may appear to be a black hole into which money disappears and from which little light is received. But to a PM, a project is a set of dependent and independent tasks.

It’s not surprising that if we ask a PM to estimate a project, the reflex reaction will be to create a WBS based upon tasks—it simply reflects how a PM understands a project. The identification and tracking of tasks is one of the most important responsibilities of the PM and, after all, at some point the PM will have to construct a detailed plan to schedule, execute, and track these tasks.

Lastly, most PMs would feel uncertain and uncomfortable defending an estimate if they were not allowed to identify the detailed tasks that went into the estimate. However, comfort notwithstanding, there are many situations where this level of detail is simply not warranted and it may imply a degree of precision that is just not there.
Measuring Mountains

There are two ways we could estimate the size of a mountain. One way would be to walk around the mountain sizing up all the boulders that went into the structure. As shown in Figure 1, there may be big boulders, small boulders, and even little pebbles. The volume and mass of the mountain can be calculated by adding together all the boulders. How the boulders collectively rest against each other would give us the height of the mountain.

An alternative to this method would be to triangulate the mountain and take several global measurements from different places, as shown in Figure 2. Or perhaps we could compare this mountain against other mountains of known size. Given just a few measurements of this type, and a couple of formulae to convert the angles and distances into a height, to translate the height into the volume of a cone, and using the density of rock to determine the mass, we would have the key metrics that define the mountain.

The Same Only Faster

I have often seen this “boulder counting” at companies where PMs work to identify and sum their tasks to produce their estimates. The resulting estimates might be good or they might be bad, but they certainly take a long time to produce.

At one client, a VP told me “…it takes them six weeks to come up with a lousy estimate, you’d be doing us a favor if you can get them to come up with an equally lousy estimate in, say, a couple of hours.” He was being facetious, but he was also talking about a genuine presumption of who would do them and much negotiation occurred along the lines of “…yes, but if we give it to Bill to do, it will only be two hours instead of four…” The PM lamented that he and his colleagues would spend hours and hours negotiating and debating, filling in all the pebbly cells with their tiny tasks. Then they would crank the handle and roll all the numbers up to the top to create the estimate. They would take this to the executives for approval and commitment only to be told “…uh, we don’t like this answer, give us another one…” So back to the rock pile they would tramp, to rearrange and recount the boulders and pebbles and several days later reappear with a different answer. This was sometimes repeated over and over, as the executives would say in effect “…we don’t like this rock, bring us another rock…”

At Least We’ve Got a Plan

Not that this process is entirely without merit. As one practitioner pointed out, when the estimate does finally get accepted they have obtained approval on the plan at the same time. But this exposes the true nature of counting boulders—it is a planning activity rather than an estimation activity. Planning is necessary and valuable, but the best plan is not much use if it cannot be staffed, or the company can’t afford it, or the customer won’t pay for it. The counting boulders approach
tends to put the task planning cart in front of the scope estimation horse.

Scope-based estimation uses a triangulation approach. It relies on projecting a value that represents the “size” of the system, and then uses a formula to convert this to the key outputs of a project estimate: the likely duration, effort/cost, and staffing needed to create a system of that “size.” Of course, the issue is determining the size. How do you know how big a system will be before you’ve built it? Or, given the normal timeframe in which an estimate is created, how do you know the scope of a system before you even understand what it’s supposed to do? The answer is: with difficulty.

Garbage in, Garbage out

One of the challenges of scope-based estimation is that most estimation approaches such as those used in the COCOMO model [1] or the SLIM-Estimate® tool, use an exponent of final system size as one of the primary factors in calculating the effort and schedule. This has the effect of amplifying any variance on the input size/scope. Any difference in size between expected and actual will result in a value raised to a power difference in output. However, that is a correct interpretation of reality. We can reasonably predict the effort and schedule of a project only to the extent we can reasonably figure out how big the task is.

The answer to this dilemma is simple. The defining characteristic of modern software development is uncertainty. This uncertainty is not simply present in the measurement activity—it is present in the development activity. When we begin a software project we don’t know in detail what it is we need to do. Our primary job as software developers is to figure this out. Once that activity is out of the way, building the system is usually pretty straightforward. Since it is inherent in the job, any legitimate estimation process must embrace and model this uncertainty, otherwise it is behaving dishonestly. Maybe we can’t measure how big the system is, but perhaps we can measure how much we can’t measure how big the system is.

The best estimation approaches and tools do this; they expect the definition of the system scope to be given in terms of a likely range. They then convert that range into a probability profile of success on the output. Variance in scope becomes risk in results. In fact, we also need to admit that often we don’t know in advance exactly how effective we will be in building this system of uncertain size. Recognizing this means we have another source of uncertainty the estimation process must accommodate. But this is the only honest way to estimate—if we really don’t know how big the thing is, and we don’t know how good we are, we really can’t know how long it will take us to build it. We just need to use a process that quantifies this uncertainty and allows us to make informed decisions based on it.

29,002 Feet

When Andrew Waugh renamed Peak XV in honor of his predecessor as Survey-General of India, he also decided that he needed to “adjust” the measured height of the highest point on Earth. He quite correctly thought that people would consider his measurements to be inaccurate if they happened to land on a multiple of 1,000. Then, as the story goes, Sir Andrew went and added a couple of extra feet to the height of the mountain because of this presumption. And so for a long time the height of Mount Everest was quoted as 29,022 feet, the extra “2” clearly demonstrating the exquisite precision of the surveyor’s art and the care to which Sir Andrew went to ensure the exactness of this measurement.

But notice, he didn’t try to count all the boulders in the mountain.

Reference


Phillip G. Armour (armour@corvusintl.com) is a senior consultant at Corvus International Inc., Deer Park, IL, and a research director in the Center for Software Development Innovation at Number Six Software, Vienna, VA.

© 2006 ACM 0001-0782/06/0100 $5.00