Courtesy of:
Scitor Corporation
256 Gibraltar Drive • Sunnyvale, CA • 94089
800/533-9876

Estimating Task Durations

A Look into the Methodologies and Psychology of Estimating

An entire article devoted to a discussion on the duration of tasks? Is he kidding, you ask? But think a bit about the relative importance of task durations. A project schedule is the result of the aggregation of all of the task durations. If the durations lack validity, so does the project schedule. Fidelity in task duration estimating is essential to the development of a wholesome project schedule. And such fidelity can only be achieved via a structured and consistent approach toward establishing task durations.

How Long Does It Take To Catch A Fish?

Here's a good question. How long does it take to catch a fish? Ridiculous, you say. One can't estimate the time to catch a fish. It could be just after you cast a line in the water. It might be never or anywhere in between. As ridiculous as this sounds, that is just the feeling that goes through our minds when we are asked to estimate the duration for a task. Our first thought is "how the h… should I know". But, we can't get away with this. So we dig in and take a scientific stab at the task duration.

First, we come up with a "most likely" estimate of the duration. This is the time that we feel that it would take about 50% of the times that we were to execute the task. But, we're not comfortable with a 50% confidence factor. So we add some time that we feel that we could support about 90% of the time.

Next, we think about what we will need to start the task, including what kinds of conditions are required. If we are concerned that we will not have everything that we need to start the task, we add some more time to the task estimate (even though these issues do not impact upon the actual time to execute the task itself).

Then there is the "collection factor". When a group of tasks come together, we tend to add some more safety margin, to allow for one of the tasks to slip. Similarly, we note that there is a tendency to "lose time" between tasks. I call this the 5+5=13 rule. Two tasks, each estimated at five days, performed in series, will take 13 days because we lose three days between the completion of the first task and the start of the second task.

So what do we do? We compensate for all of these factors that are external to the immediate task, by adding time to the task estimate, itself.

Finally, everyone knows that the total duration will not be accepted. They expect to be pushed for a 20% reduction, so they add 25% to the already inflated estimate.

What Does the Task Duration Really Represent?

If we assign task durations as described above, do we really know what the expected task duration is? Certainly there is justification for all of the above-mentioned items. However, most of them have nothing to do with the actual time that we need to perform the task.

Furthermore, even the estimate of the actual task duration can take several paths. For instance, here are several approaches to estimating task durations:

Elapsed Time vs. Working Time – We feel that it will take five days to actually perform the work. But we know that we will not be working on the task without interruption. So we set the task duration at ten days, to allow for the elapsed time that we expect to occur.

Task Time vs. Resource Time – We estimate that the task will take 80 hours to perform. Is this 80 hours by two people, producing an elapsed time of five days? Or is it 80 hours for one person, working half time, producing an elapsed time of 20 days?

Interface Losses & Delays – We noted above that we could expect some loss of time between tasks and when multiple tasks converge. Shall we incorporate these expected losses into the tasks themselves, or set up dummy tasks to allow for these delays?

By the way, with any of the CPM tools, it is possible to set a lag between the end of one task to the start of a successor. For instance, to add three days between Task A and Task B, we would define the link between these two tasks as "FS3". Task B can start 3 days after Task A finishes. In reality, the start of Task B is not actually delayed. It is just the schedule that will reflect the time allowance that has been inserted.

Theoretical Duration vs. Experience – Here's a situation that always frustrates me. I have a task that I have performed several times. Each time that I estimate how long it should take, I come up with 20 days. I just know that I can do it in 20 days. Yet, each time that I perform the task, it takes about 50% longer than the 20 days. Each time there is a different reason for the delay. Nevertheless, I average 30 days to do the job. Now, what do I do? Do I use an estimate of 20 days – the duration that I feel to be most correct? Or do I use an estimate of 30 days – based on past experience? I am justified to use the 20-day estimate. The task should be completed in 20 days and this is what we should use as a target. But, if our experience tells us to expect 30 days, aren't we deceiving the team by saying that we expect it to be done in 20 days? And, if we use the 30-day

Scitor Corporation Enterprise Project & Process Management Solution Providers www.scitor.com

estimate, will we end up taking the 30 days, because that is the time available? Is there a right answer?

Here are a few traps to note:

Be careful not to improperly use averaging. For instance, we would not want to average performance on parallel paths. Let's say that we have Tasks A, B, C & D, each estimated to take 20 days. A, B & C actually take 15 days each. Task D actually takes 35 days. While the average still works out to 20 days, the actual duration for the path (for the four parallel tasks) is 35 days.

For another example, we look at two serial tasks, each estimated to take ten days. Task A gets done in 8 days. Task B takes 12 days. The chain probably took 22 days (rather than 20) because Task B didn't start until the 11th day. (Harvey's Law #121: A delay in one step is passed on to the next step. An advance made in one step is usually wasted).

Skill Levels, Learning Curves & Priorities – How do we handle potential performance modifiers? Do we add time to the duration estimate because we expect that there will be additional time and effort needed to do the task the first time (learning curve)? Should the duration be adjusted based on skill level of the resources expected to be assigned? Do we actually have an index of skill level? And what if the resources change?

Does a higher priority task or project get done faster because of the pressure and attention? These are all things that can impact upon the task duration. But there rarely is a set of guidelines in place to help us with the estimating and to aid in achieving consistency across the project and the team.

PERT Method – This technique provides for a quantitative method of considering uncertainty or risk. It calls for the use of three time estimates for each task. These are called optimistic, most likely and pessimistic. The most likely is the duration that can be expected 50% of the time. The optimistic is the shortest reasonable duration, attainable about 10% of the time. The pessimistic is the longest reasonable duration, also with about a 10% probability. In the PERT method, a PERT duration is calculated, usually based on the formula: $(a + 4b + c) / 6$, where "b" is the most likely.

Although it may appear that the PERT method takes a great deal of additional effort, the reverse is really true. In reality, we tend to go through the process of thinking of the possible range of estimates, based on perceived risk and uncertainty. But then, after mentally deriving a single duration, we fail to capture the information that went into the estimate.

Note: A special feature in Scitor Project Scheduler allows the user to vary the weighting of the three estimates. Increasing the weight of the pessimistic estimate allows the user to add more margin to the estimates. Placing all of the *weight on either the pessimistic or optimistic values allows the user to calculate the extreme dates to the project – the shortest possible and the longest possible.*

There are also a few special schedule risk software products available that can perform a statistical analysis, providing a calculated probability of meeting any project end date.

Delphi Method – This decision-aiding technique is rarely employed in determining task durations, but could be applied if desired. It calls for each member of the team to offer their own estimate to the group. Estimates at the extremes (shortest/longest) are defended by the estimator, which often introduces issues that were not considered by the others. Based on the new information, the team votes again (re-estimates). The process is repeated until there is a reasonable consensus and comfort with the task duration.

The Psychology of Task Durations

There is a self-fulfilling prophecy regarding performance of tasks within planned durations. A task is hardly ever completed ahead of schedule. There are several reasons for this. We can demonstrate these using an illustration of a task that has a 50-50 chance of being completed in five days, but has been scheduled for ten days to allow for uncertainty, risk, emergency diversions, etc.

First, there is Parkinson's Law: *"Work expands to fill the time available for the work"*. Work on the task has commenced on schedule and is essentially completed within the first five days. But, because ten days have been allocated for the task, the performer spends the next five days "fine tuning" the deliverable. This is a natural work ethic of most people. We reach 98% completion on our task and, if additional time is available, we attempt to refine it until a delivery deadline is reached.

Second, is procrastination. We are able to start the task as scheduled. But, because there are ten days allocated, and we know that we only need five days, we wait a week to start the task. Now, of course, the contingency has been exhausted before the task has been started, and the potential for a schedule overrun has been increased. But, even if there are no problems, the five-day task has taken ten days.

Less obvious are the subtle motivators to avoid "early" completion of tasks. If we estimated ten days and complete the task in five days, we might be criticized for "padding" the estimate, even though the extra five days was a legitimate allowance for uncertainty. Or, we might be under increased pressure to shorten duration estimates in the future. There rarely is a reward for finishing tasks early – only demerits for running over. So where is the motivation to do the task in five days?

Harvey's Law #134: The time to complete a task will almost always take a minimum of the allocated time, and probably more. If pressure is to be maintained to minimize the time spent on tasks, it is advantageous to move most contingency out of the individual tasks and allow for the contingency in other ways.

A method that is gaining popular support is the concept of "shared contingency" (my term) that has been publicized by Eliahu Goldratt (in his book; "Critical Chain"). This method is supported by Scitor Project Scheduler.

Practical Time Estimating

Recognizing all of the possibilities for distorted or padded time estimates, how can we allow for all of the perturbations that are likely to impact upon the schedule, without masking the true duration estimate for the task? Certainly, if we do not allow for uncertainty, by adding contingency, we risk a high potential of running late and missing deadlines. However, if we bury the contingency in the individual task estimates, we almost assure that the work will slip to fill the time available.

It is this dilemma that motivated the concepts of Shared Contingency. Use of the various shared contingency conventions is one way of addressing many of the issues raised above. It is also feasible to deal with some of these issues using traditional CPM methods and tools. Here are a few illustrations:

Example 1 – Task should be completed in 20 days, but need to allow 30 days in schedule based on past experience. Enter a duration of 20 days. Create a dummy task for contingency, with duration of 10 days.

Example 2 – Lump the entire contingency for a logical group of tasks in a shared contingency dummy task. Using Goldratt's Critical Chain PM philosophy, add up the contingencies and cut in half for the dummy (buffer) task.

Example 3 – Use finish-to-start (FS) links with a lag duration to incorporate time for delays between tasks.

Example 4 – Freely impose Finish-No-Later-Than (FNLT) dates to drive earlier completions. Set FNLT dates equal to the Early Finish dates for tasks that you do not want to let slip.

More important than all of the above is the need to develop consistency in estimating task durations. There should be a blanket policy for contingency. At least that way everyone knows the basis for the estimate. Standard guidelines for task duration estimating should be established by the project's function for universal use.

The application of the guidelines should consider the key factors in achieving project success. If getting the job done as fast as possible is a key objective, then contingencies should be minimized and identified. If protecting the firm from delay penalties is a key issue, then contingency allowances play a larger role.

Flexibility, within standardized guidelines, together with notation of and communication of the basis for the estimates, will help reduce the potential for poor estimating and scheduling. Hey! Nobody said it was going to be easy.

Harvey A. Levine, with 38 years of service to the project management industry, is founder of **The Project Knowledge Group**, a consulting firm specializing in PM training, PM software selection, evaluation & implementation, and PM using microcomputers.

He has implemented or enhanced the project management capabilities of numerous firms, often combined with the selection or implementation of computerized project management tools. Mr. Levine is considered the leading consultant to the project management software industry and is recognized as the leading expert in tools for project management.

He has been an Adjunct Professor of Project Management at Rensselaer Polytechnic Institute and Boston University. And has conducted numerous project management public seminars for ASCE, AMA, IBM, and PMI.

Mr. Levine is the author of the book "Project Management using Microcomputers", and has been published extensively in other books, periodicals and videos.

Mr. Levine is a past president of the Project Management Institute and the recipient of *PMI's 1989 Distinguished Contribution to Project Management* award. Recently, he was recently elected as a *Fellow of PMI*.

Mr. Levine has offices in Saratoga Springs, NY and San Diego, CA and can be contacted via e-mail at: LevineHarv@cs.com

Scitor Corporation
Enterprise Process & Project Management Solution Providers
www.scitor.com