Making Sense of Schedule Risk Analysis

(Presentation made to Australian Institute of Project Management Project Controls CoP, November 2013.)

Introduction

I am going to assume you all know CPM scheduling but will start from first principles as far as probability and statistics are concerned. I apologize in advance if I am teaching grandmother to suck eggs; bear with me for a while.

The Nature of Uncertainty (It's all about YOU!)

Perhaps I should start by saying that while we are stuck with the term "risk analysis" for what I am about to talk about, I prefer the term "uncertainty" to "risk." It is neutral, whereas "risk" has negative connotations. "Risk" is also often taken to refer to discrete events which either materialize or not, whereas uncertainty is more general.

Uncertainty is often about the future, but it needn't be. For example, consider the question:

> How long is the Suez Canal?

This has a very precise answer, yet probably you do not know. I did not know when I started to prepare this paper. If I had been forced to come up with a single-value answer, it would almost certainly have been wrong. But if I were allowed to specify a range I would have had a good chance of getting it right. I might have said for example "between 50 and 200 miles." (Say 80 to 320 km).

According to Wikipedia, the actual length is 193.30 km (120.11 miles). That is a very precise figure, hardly uncertain at all.

> *My point is that the uncertainty is not inherent in the reality of the Suez Canal but in our state of knowledge about it.*

Arguably this is true of future events as well.

Before I go further, I should mention that uncertainty is often divided into two categories:

- Aleatoric uncertainty, and
- Epistemic uncertainty.

There seem to be varying definitions of these terms, but here are typical ones:

- **Aleatoric uncertainty** is uncertainty that comes from a random process. Flipping a coin and predicting either heads or tails is aleatoric uncertainty.
- **Epistemic uncertainty** is uncertainty that comes from lack of knowledge. This lack of knowledge comes from many sources. Inadequate understanding of the underlying processes, incomplete

> knowledge of the phenomena, or imprecise evaluation of the related characteristics are common sources of epistemic uncertainty.

Whether the first kind of uncertainty exists in the real world is in my view questionable. I would argue that the randomness described above is in *our model* of the coin flip, not in the coin flip itself.

Laplace believed this:

> *"Given for one instant an intelligence which could comprehend all the forces by which nature is animated and the respective positions of the beings which compose it, if moreover this intelligence were vast enough to submit these data to analysis, it would embrace in the same formula both the movements of the largest bodies in the universe and those of the lightest atom; to it* ***nothing would be uncertain****, and the future as the past would be present to its eyes."* [My emphasis.]
>
> Pierre-Simon Laplace, Theorie Analytique de Probabilites (1812-1820)

Of course Laplace did not know about Heisenberg, let alone chaos theory. But I know at least one physicist who thinks even quantum uncertainty is epistemic.

Here is another interesting definition of aleatoric, from Wikipedia:

- ***Aleatoric uncertainty****, aka statistical uncertainty, which is representative of unknowns that differ each time we run the same experiment. For an example of simulating the take-off of an airplane, even if we could exactly control the wind speeds along the runway, if we let 10 planes of the same make start, their trajectories would still differ due to fabrication differences. Similarly, if all we knew is that the average wind speed is the same, letting the same plane start 10 times would still yield different trajectories* ***because we do not know the exact wind speed at every point of the runway, only its average****. Aleatoric uncertainties are therefore something an experimenter cannot do anything about: they exist, and they cannot be suppressed by more accurate measurements.* [My emphasis.]

This seems to contradict the idea that aleatoric uncertainty is irreducible in principle, since the reasons given for our lack of knowledge are practical rather than theoretical. (Why assume that we know the average wind speed but not the speed at every point? In fact, how would we know the average *without* knowing it at every point?)

This leads me to think that the distinction is not so much about the reality as about how we choose to model reality. We *choose* to model a coin toss as truly random because we cannot contemplate all the data we would need to do otherwise.

Whatever we think about this argument, it is pretty safe to say that most of the uncertainty encountered by project managers in predicting the course of their projects is of the epistemic variety. That is, we could in principle reduce or eliminate it, but it is either impractical or too costly to do so. In other words *it is not about the variable in question, it is about our state of knowledge.*

Or to quote Laplace again:

> *"Chance is merely a measure of our ignorance."*
>
> Pierre-Simon Laplace

I will come back to the implication of this later, but first we need to talk about how we measure uncertainty.

Probability Distributions

The simplest probability distributions are discrete, that is to say they enumerate the probability of each of a set of discrete possibilities. So, for example in the toss of a fair coin there is a 50% probability that the result will be a head and 50% that it will be a tail.

> *[But note the word "fair." How do we know when a coin is fair? How do we define fair without the tautology of saying that it has a 50/50 chance of either outcome? With a real coin, how many consecutive heads would we need before we start to question whether it really is a fair coin? Isn't that a subjective decision based maybe on who gave us the coin? Do fair coins exist or are they just an abstraction like the frictionless pulleys and weightless strings of our schoolboy physics days?]*

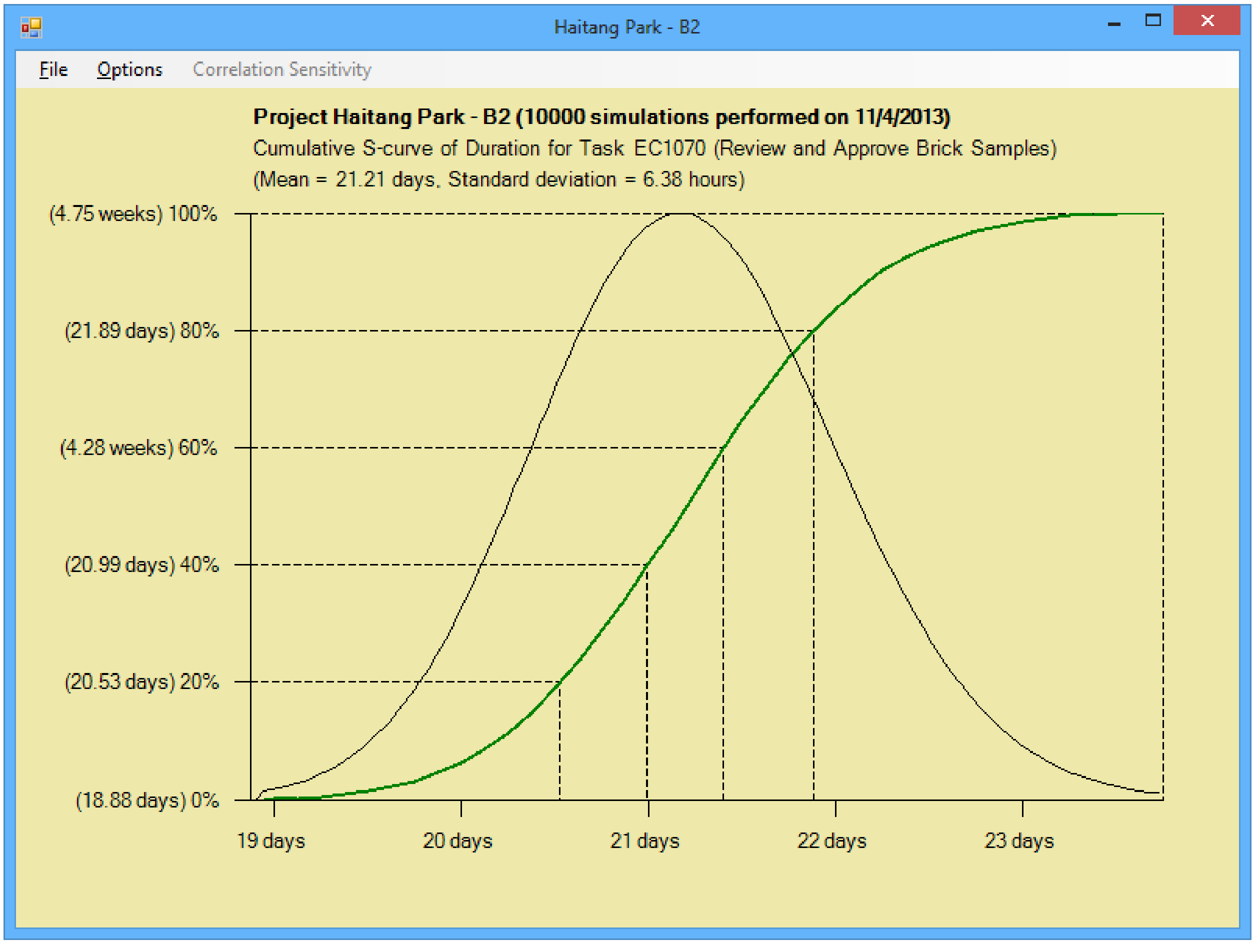
Most distributions are continuous, meaning that the actual result may take any value on a continuum. (In project management we tend to measure to the nearest minute even though in practice we care more about days and weeks, so for practical purposes this means that the distributions we are concerned with are continuous.)

Note that since there are an infinite number of possible values, the actual probability of any particular value is infinitesimal. A continuous probability distribution (or *frequency function*) therefore represents the *probability density*, or probability per unit of measure of the variable.

The grey curve below is an example of a frequency function.

We can get back to more familiar ground by looking at the *cumulative distribution function*, which is the integral of the frequency function, i.e. the area under the curve. Each point represents the probability that the actual result will be less than a certain amount. By definition, the CDF is monotonic non-decreasing, starting at zero and going to 100%.

The green curve below is an example of a cumulative distribution function.



Subjective Probabilities

I have dwelt on the nature of uncertainty, and in particular the epistemic variety, because a common objection to risk analysis is that the inputs are generally subjective. This is because statistics is generally taught in terms of experiments like coin tosses which can be repeated any number of times. And we use the aleatoric model which says we do not know or cannot process any of the detail which actually makes each individual toss different.

Tasks in projects are often (some would say always) one-off events which have never been done before. Does it make sense to talk about the probability that a one-off event will take longer than a certain amount of time?

I would say yes. For one thing, we have no choice. Even if we stick with a single-point estimate we are in effect saying something like “there’s a 50% chance that the task will take no longer than x.”

For another, we do it all the time. At least gamblers do. They bet on elections, horse races, and all sorts of other one-off events. Bets are interesting because the bookie sets the odds based upon the bets he receives, which means that for everyone who thinks the chance of a particular horse winning is better than the odds offered, there are others who think the opposite. A similar argument can be made about the stock market.

Again, it is all in our minds. In other words it is *always* subjective.

One more quote, from an excellent book:

> *"He saw no fundamental irony in his position: Because he believed he did not have enough data to estimate a range, he had to estimate a point."*
>
> Douglas Hubbard, "The Failure of Risk Management."

It is precisely *because* we do not know as much as we'd like about something that we *should* estimate a range rather than a point. In any case, a point estimate is also subjective, and what's more it is almost bound to be wrong.

Measures of Central Tendency

It is often desirable – though as we will see it can also be highly dangerous -- to summarize a probability distribution with a single number. There are several candidates for this:

- The **median** is the point at which the CDF is 50%. There is a 50% chance that the value will be less than the median and a 50% chance that it will be greater.
- The **mode** is the most likely value, which is the highest point on the frequency function.
- The **mean**, or expected value, is what most people call the average, is simply the weighted average of the frequency function:

$$\int x\, f(x)\, dx$$

Which measure of central tendency is appropriate depends on the purpose to which it is to be put (and sometimes none of them are appropriate).

More often than not we use the mean. The main advantage of using the mean is that you can add means together, i.e.:

$$\text{Mean}\ (x + y) = \text{mean}\ (x) + \text{mean}\ (y)$$

This is not true of the median or mode, except in the case of symmetrical distributions where all three measures are the same. Note that this is a practical advantage rather than a theoretical one.

However, there are many situations where using the mean (or either of the other measures) is not appropriate. In general, as Sam Savage says in another excellent book:

> *"The mean of the function is not the function of the mean."*
>
> Sam Savage, "The Flaw of Averages."

For example the expected value of x^2 is not the square of the expected value of x.

Critical path analysis happens to be a very good example of this. *Unbiased estimates of the durations of individual tasks do not result in unbiased estimates of schedule dates.* This is because of a phenomenon called merge bias.

Merge Bias

Merge bias occurs whenever two or more paths converge in a network and the uncertainty about their durations is such that any of them might turn out to be critical. Consider a task which has just two predecessors which run in parallel. Suppose that each could take anything from 1 to 6 days with equal probability. And suppose further that they will not take fractions of a day. This may sound rather contrived, but I am doing it so we can simulate them with a pair of dice and not get involved in a lot of calculus.

The expected duration of the individual tasks is of course 3.5. But what is the expected time for them **both** to be complete? Let's consider the dice to find out. The following table shows all 36 possible outcomes of throwing two dice.

	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	2	3	4	5	6
3	3	3	3	4	5	6
4	4	4	4	4	5	6
5	5	5	5	5	5	6
6	6	6	6	6	6	6

Of the 36 possibilities only one has 1 as the maximum while no less than 11 have 6 as the maximum. So already with just two parallel paths we can see that the time it takes for both of the tasks to be complete is likely to be longer than the deterministic estimate of 3.5 days. If we do the math we find that the expected value is actually almost 4.5 days.

We cannot visualize this easily for more dimensions, corresponding to more parallel tasks. With 3 parallel tasks, the expected value is almost 5 days, and with 5 tasks it is about 5.4 days. And with 10 it is 5.8 days. Clearly it will converge on 6 days for a large number of such paths. [In the live presentation I use an Excel spreadsheet to calculate the values for larger numbers of merging paths.]

Another way to look at this is to restate Murphy's Law, less pessimistically than Murphy:

> *"If there are <u>many</u> things which could go wrong then <u>at least one of them</u> probably will."*

Finally, I should point out that the bias associated with a single merge point is rarely as extreme in real projects as in the above example. On the other hand, real projects have many merge points and the bias builds up throughout the project, making it progressively harder to meet milestone dates as the project progresses. Also note that there is no countervailing tendency to create any bias in the opposite direction.

In my view a major reason why projects so often seem to run late is that the plan was never realistic in the first place. I am not alone in this belief:

> *“Late chaotic projects are likely to be much later than the project manager thinks -- project completion isn’t three weeks away, it’s six months away. ……., but that’s not a result of Brooks’ Law. It’s a result of underestimating the project in the first place."*
>
> (Steve McConnell, “Brooks' Law Repealed," IEEE Software, vol. 16, no. 6, Nov/Dec, 1999. Brooks’ Law states that putting more resources onto a project when it is running late will make it take even longer.)

And failure to take uncertainty – and therefore merge bias – into account is a major reason why project plans are unrealistic.

Merge bias is probably the most important reason to do risk analysis. It is also a major reason to do risk analysis on a detailed network. If you use a summary network, many of the merge points will not be represented and the results will be overly optimistic. It is also a reason NOT to use PERT.

PERT

PERT was an early attempt to deal with uncertainty in project networks. It was loosely based upon the assumption that durations follow a subset of the beta family of distributions. The subset was selected so that the user would make a three-point estimate of the duration, and the mean and standard deviation would be approximated by:

Mean = (optimistic + 4 * most likely + pessimistic) / 6

Standard Deviation = (pessimistic – optimistic) / 6

Note that these are only approximations, but this is not the real problem with PERT. The problem is that it then calculates the critical path using the means as estimated above, and assumes that this is the only possible critical path. With uncertainty associated with all the task durations, there are usually a number of paths which could be critical.

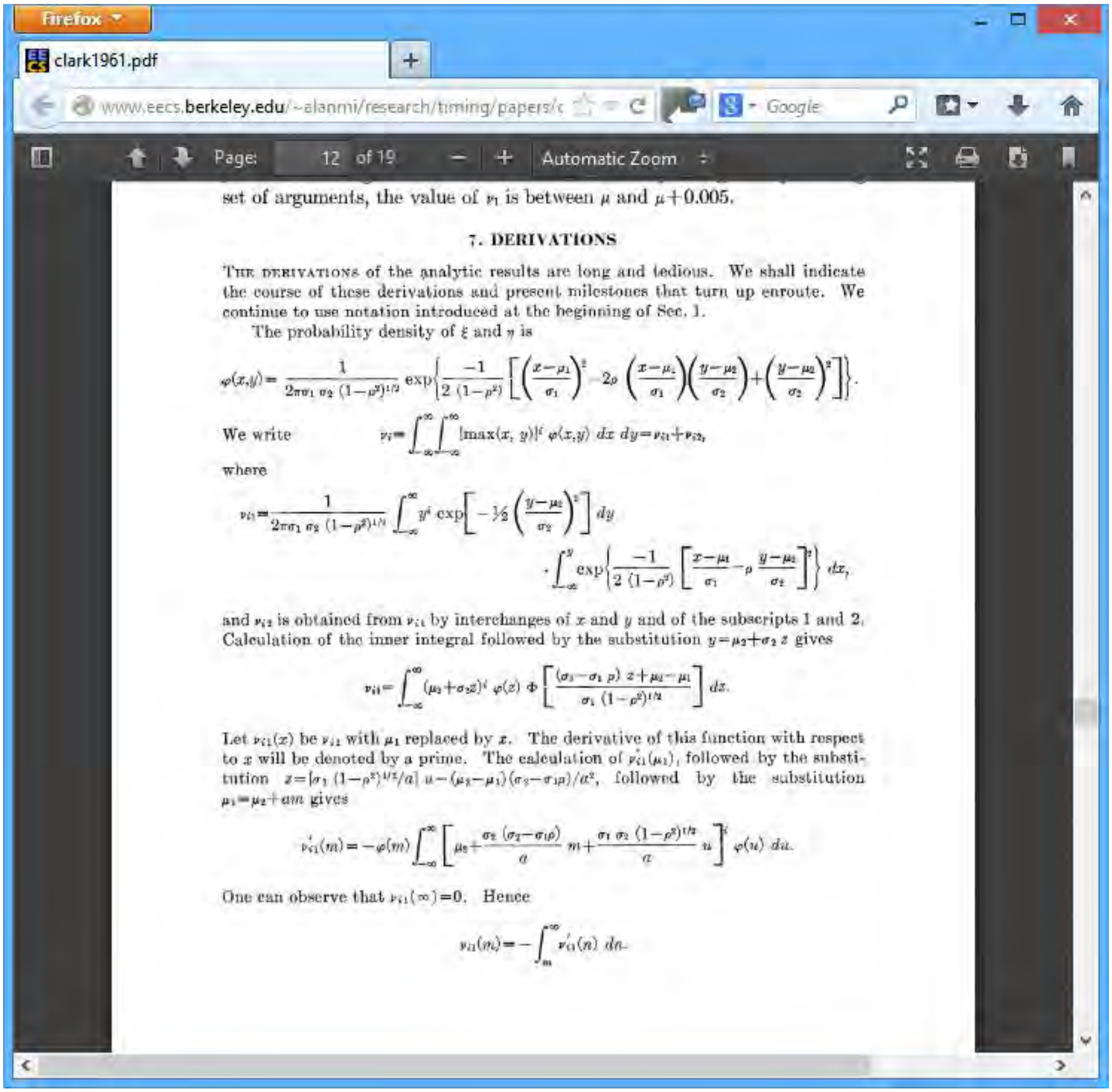
If we apply this to the dice example above, we get a mean of 3.5 (regardless of how many parallel paths we have) and a standard deviation of 0.84. Which means we would estimate the 80th percentile (sometimes called the P80 point) at about 4.4. Note that even with just two parallel paths, this is not even the P50 point! The more such paths, the more biased the PERT result is.

The authors of the original PERT paper in 1959 recognized this problem, saying that “PERT always produces estimates which are too optimistic.” They justified the methodology on the basis that there was insufficient computing power available at the time to do it properly. This is no longer true.

Other Attempts at Analytical Solutions

One of the authors of the original paper tried to improve on it, publishing another paper in 1961. ("The Greatest of a Finite Set of Random Variables," Charles E. Clark, Operations Research, 1961.) The use of the words "finite set" in the title is a bit misleading, because he restricted himself to 2, saying that for more than 2 variables (equivalent to more than 2 merging paths) the problem "became cumbersome."

Nevertheless, even for 2 variables the math is nothing short of heroic, as you can see from the opening page:



set of arguments, the value of ν_1 is between μ and $\mu+0.005$.

7. DERIVATIONS

THE DERIVATIONS of the analytic results are long and tedious. We shall indicate the course of these derivations and present milestones that turn up enroute. We continue to use notation introduced at the beginning of Sec. 1.

The probability density of ξ and η is

$$\varphi(x,y)=\frac{1}{2\pi\sigma_1\sigma_2(1-\rho^2)^{1/2}}\exp\left\{\frac{-1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_1}{\sigma_1}\right)^2-2\rho\left(\frac{x-\mu_1}{\sigma_1}\right)\left(\frac{y-\mu_2}{\sigma_2}\right)+\left(\frac{y-\mu_2}{\sigma_2}\right)^2\right]\right\}.$$

We write

$$\nu_i=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[\max(x,y)]^i\,\varphi(x,y)\,dx\,dy=\nu_{i1}+\nu_{i2},$$

where

$$\nu_{i1}=\frac{1}{2\pi\sigma_1\sigma_2(1-\rho^2)^{1/2}}\int_{-\infty}^{\infty}y^i\exp\left[-\tfrac{1}{2}\left(\frac{y-\mu_2}{\sigma_2}\right)^2\right]dy\cdot\int_{-\infty}^{y}\exp\left\{\frac{-1}{2(1-\rho^2)}\left[\frac{x-\mu_1}{\sigma_1}-\rho\,\frac{y-\mu_2}{\sigma_2}\right]^2\right\}dx,$$

and ν_{i2} is obtained from ν_{i1} by interchanges of x and y and of the subscripts 1 and 2. Calculation of the inner integral followed by the substitution $y=\mu_2+\sigma_2 z$ gives

$$\nu_{i1}=\int_{-\infty}^{\infty}(\mu_2+\sigma_2 z)^i\,\varphi(z)\,\Phi\left[\frac{(\sigma_2-\sigma_1\rho)\,z+\mu_2-\mu_1}{\sigma_1(1-\rho^2)^{1/2}}\right]dz.$$

Let $\nu_{i1}(x)$ be ν_{i1} with μ_1 replaced by x. The derivative of this function with respect to x will be denoted by a prime. The calculation of $\nu'_{i1}(\mu_1)$, followed by the substitution $z=[\sigma_1(1-\rho^2)^{1/2}/a]\,u-(\mu_2-\mu_1)(\sigma_2-\sigma_1\rho)/a^2$, followed by the substitution $\mu_1=\mu_2+am$ gives

$$\nu'_{i1}(m)=-\varphi(m)\int_{-\infty}^{\infty}\left[\mu_2+\frac{\sigma_2(\sigma_2-\sigma_1\rho)}{a}\,m+\frac{\sigma_1\sigma_2(1-\rho^2)^{1/2}}{a}\,u\right]^i\varphi(u)\,du.$$

One can observe that $\nu_{i1}(\infty)=0$. Hence

$$\nu_{i1}(m)=-\int_{m}^{\infty}\nu'_{i1}(n)\,dn.$$

He also had to assume that the incoming distributions were normal. (Note the conflict with the original PERT assumption.) The problem is that the outcome is decidedly not normal, so even if the normal assumption applies for the first merge point, it most certainly does not for subsequent ones.

Monte Carlo Simulation

This is why it has been generally accepted since about the mid- seventies that the only way to deal with uncertainty in project networks is by Monte Carlo simulation. So what is Monte Carlo simulation?

At its simplest, and in the context of project schedules, it involves:

1. Sampling task durations from user-specified distributions;
2. Performing CPM calculations as if these were the actual durations;
3. Repeating this many (typically thousands) of times, each time with a different set of samples;
4. Summarizing the results as distributions of the outputs of interest e.g. the project finish date.

[In the presentation, I demonstrate Monte Carlo simulation with live software in the context of the dice example, in the context of a project schedule, and in the context of estimating the value of pi.]

The lessons learned are:

- That Monte Carlo simulation works, giving answers in these simple cases which match the known theoretical answers.
- But that it works only approximately, relying on large numbers of trials to get good results.

Which brings me to the need for speed. Many software products are too slow, tempting users either to do too few trials (and thus to get unreliable results) or to use a summary network (and thus to get biased results) or both.

Full Monte is at least 10 times faster than competing products. In one case, on a very large project, it was 2,700 times faster. In other cases, other software has been aborted after 24 hours on tests Full Monte completed in minutes.

Choosing Distributions

A common objection to schedule risk analysis is how to determine the correct distributions. We have already dealt with this to some extent. Remember:

- There is no “correct” distribution. The distribution is an expression of our ignorance, and so is necessarily subjective.
- We have no choice but to make estimates, and range estimates are easier and more reliable than point estimates.

Having said that, we still have choices to make. Full Monte supports various families of distributions:

- Normal
- LogNormal
- Beta (aka PERT Beta)
- Triangular
- Uniform

Except for the last one (which I would not recommend except in very special circumstances) it makes little difference which you choose.

Dan Trietsch of the American University of Armenia offers some empirical evidence that the lognormal distribution is often appropriate. (See Further Reading below for reference.)

If you want control over the skew of the distribution, then use beta or triangular.

But the choice won't make a lot of difference. More important to get the range right. To quote again from Sam Savage's book "The Flaw of Averages":

> *"In the land of averages the man with the wrong distribution is King."*

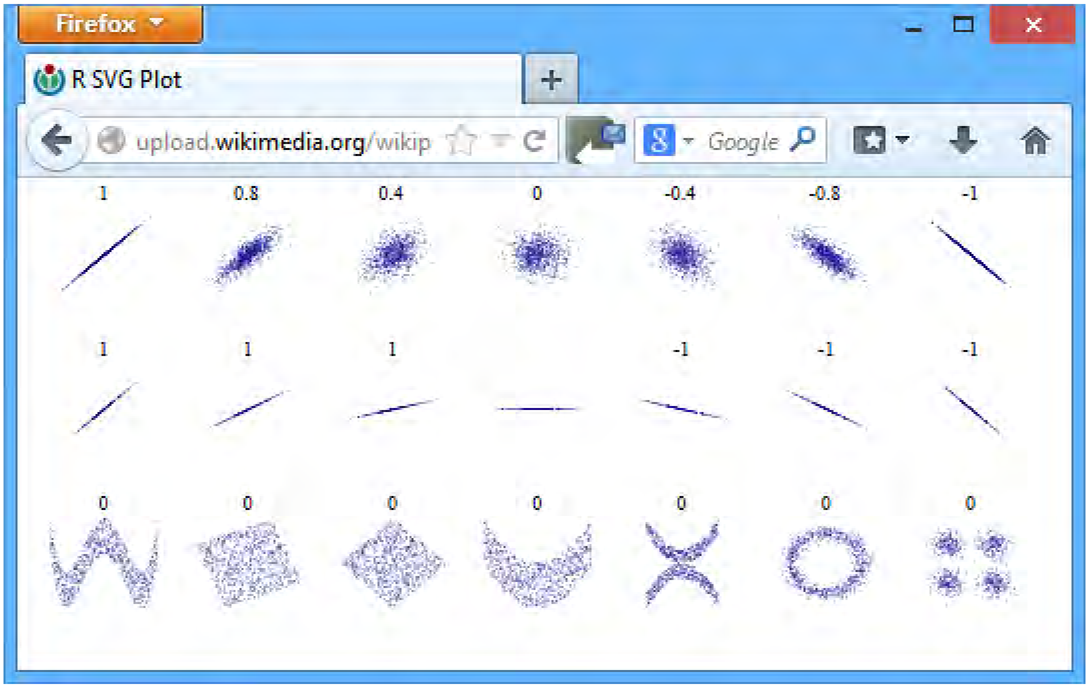
Correlations

Correlations are the cause of quite a lot of confusion. Correlation is a measure of the extent to which two random variables (in our case, typically task durations or maybe the duration and cost of a particular task) tend to vary together. Correlation coefficients vary from -1 to +1.

If the correlation coefficient is zero, the two variables are said to be independent. The correlation between two throws of a die is zero, at least in our aleatoric model of it.

On the other hand, the correlation between the weather on two consecutive days is positive.

There are actually several measures of correlation, but mostly we mean the one due to Pearson. This illustration from Wikipedia is a good guide to what it means, and what it does not mean:



Note that it is NOT about the slope of the relationship.

Also note that a zero correlation does NOT mean that the variables are unrelated, though in practice we generally take it to mean that. (The last line of scatter diagrams above make the point but are not likely to bother us in real life.)

Correlations should be used in Monte Carlo simulation when one has reason to think two variables are related in a way which makes an above-average value for one tend to coincide with an above-average value for the other. For example, two design tasks being undertaken by a new and untried subcontractor.

Of course, correlations can occur between sets of more than two variables, and these are generally represented by a correlation matrix. There are rules about what this matrix can look like. It should be pretty obvious for example that if A is perfectly correlated with B and B with C then A and C must also be perfectly correlated. The more general rules are complex and outside the scope of this presentation.

Full Monte does away with an explicit correlation matrix, and the rules that go with it, by allowing each task to be correlated with any number of outside influences (which we call "correlation sources") which are assumed to be mutually independent. This ensures the rules are observed without the user having to worry about them.

One very bad reason for using correlations, which I have heard suggested by more than one "expert" who should know better, is to avoid the consequences of the Central Limit Theorem. What they apparently mean is to avoid the fact that the sum of a number of independent random task durations will exhibit less proportional variation than each individual one. (The theorem actually relates specifically to the distribution of the mean of samples taken from a single distribution, and what it says is that this distribution tends toward the normal distribution as the sample size increases, so they are also misusing the term, but that is a minor quibble.)

But this is a real phenomenon. If one needs to overcome it, that can only mean that the individual distributions one has specified are wrong, or maybe that one just does not like the results. Pretending that they are correlated is at best a blunt instrument to correct bad estimates. (As task durations are not all additive – some tasks being done in parallel – it is by no means clear how much pretend correlation one should add. Nor is it clear what one should do if this conflicts with real correlations.)

My advice would be to use correlations sparingly if at all, and only when there is good *a priori* reason.

Simulating Calendars

One reason often given for using correlations is the weather. Tasks which require good weather are correlated for that reason. Again, this is a blunt instrument, because it assumes that the tasks occur at the same time. If one task is to be done in January and the other in March, the fact that they are both weather-dependent would not make them correlated. (Correlation coefficients between various measures of the weather on successive days tends to be around 50%, after removing predictable seasonality, but dissipate over time. After a week it is down to about 1%.)

Far better to apply randomness to the calendar itself. Upcoming versions of Full Monte allows you to specify the probability that whole or partial days or weeks can be lost. Clearly it is also seasonal, which Full Monte supports by allowing the probabilities to be specified by week number.

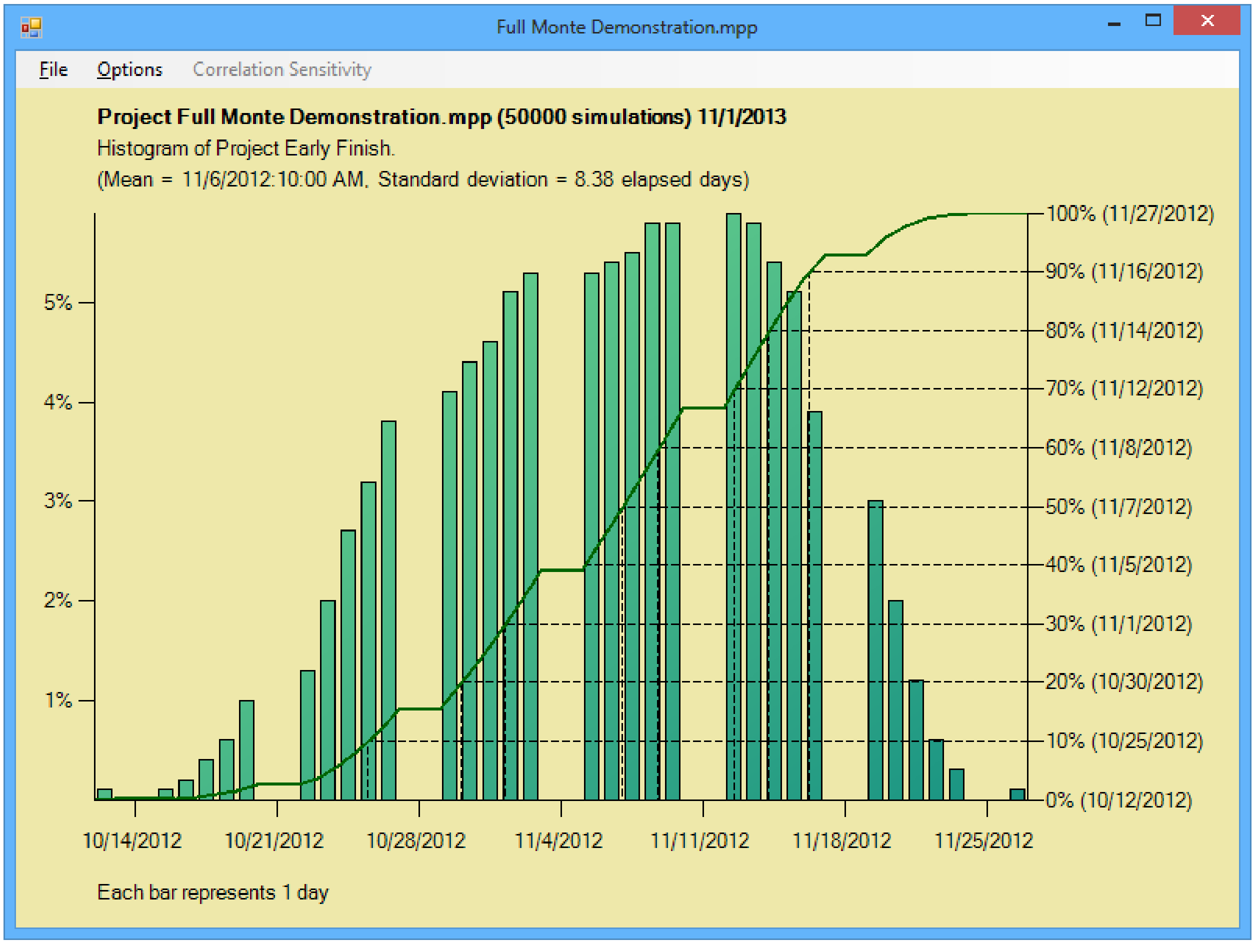
Interpretation of Results

Risk analysis typically generates a lot of results. Full Monte produces probability distributions for the following for each task:

- Early start and finish
- Late start and finish
- Free and Total Float
- Cost
- Duration

The most important of these is generally the early finish of the whole project. Full Monte presents this by default as a combined histogram and S-curve as shown below. The histogram represents the frequency function while the S-curve represents the cumulative distribution function.

Other products are similar, though some do not present results for every task.



The histogram looks pretty but the S-curve is the more useful. One can read off the probability that the project will finish by a certain date.

One can do the same for intermediate milestones, and in fact any task.

In tabular reports one can get summary information like the expected value and standard deviation of all the same variables. One issue which sometimes comes up is how dates should be averaged.

If task durations are integer numbers of days, then in a deterministic schedule the tasks will generally finish at the end of a day, say 5 pm. In the simulation, they will sometimes finish earlier and sometimes later. In the cases when they are later, they will be _much_ later, typically 8 am the next day, or even the next Monday. When we average these values in the straightforward way we will get a date/time which is somewhere around midnight on that night (or on the Saturday night if a weekend is involved).

The fact that this is during a non-work period is analogous to the fact that the expected value of a dice throw is 3.5, or that the average family has 2.4 children. The *expected* value is not necessarily a *possible* value. (And we don't therefore actually "expect" it!)

In addition to being on a non-work period, the expected date is also later than the deterministic date. (This is a spurious result, since there is no work time between the two dates, and is different from the effect of merge bias, which is real.)

But there is another way to average the data. We can calculate the average based upon work time, and then convert it to calendar time. To see the difference, consider a simple case where we did just two iterations; one resulted in a finish date of 3 pm on a Friday and the other 9 am the following Monday. (I am assuming a 5-day week working 8 till 5.)

Averaging these two values the obvious way results in an expected value of midnight on Saturday. Averaging the other way results in 4 pm on Friday. Opinions seem to differ on which is the more realistic. I prefer the second answer, but the downside is that the result can be very unstable, switching spuriously between Friday evening and Monday morning.

Other useful results from the simulation include the criticality index, i.e. the percentage of trials in which a particular task was critical. This is analogous to the critical flag in a deterministic analysis but more useful because it typically shows multiple paths which *might* become critical.

Sensitivity Analysis

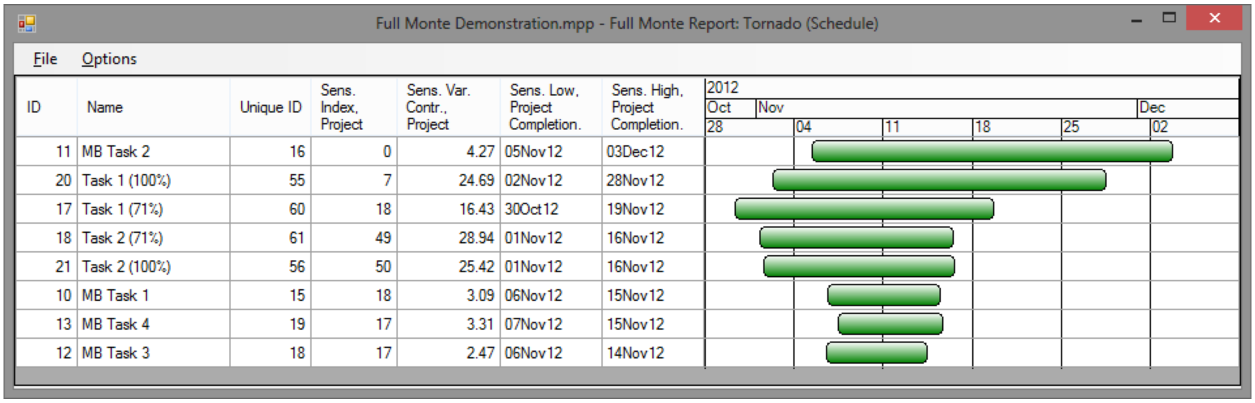
Finally, one output which can be extremely useful is sensitivity analysis. This estimates the potential effect of uncertainty about the duration of any particular task on the project completion date (or the date of any other significant event), all else being equal. It is not always considered part of risk analysis – indeed it is sometimes considered as an alternative way of dealing with uncertainty – but to do it properly it should be.

To see why, consider the phrase "all else being equal." When sensitivity analysis is done in isolation from risk analysis, this is generally taken to mean that the durations of all other tasks take their most likely values. To see how misleading this can be, consider an extreme example in which there are say 10 identical parallel tasks merging. Each can take between 4 and 8 weeks, with the most likely value of 6 weeks. If we pick one at random and do the CPM twice, one with its duration set to 4 weeks and one with it set to 8 weeks and with all the others set to 6 weeks, it will seem that this task alone could affect the end date by 2 weeks.

In reality, however, one of the other nine tasks is almost bound to take close to 6 weeks, meaning that the selected task on its own has very little effect on the end date.

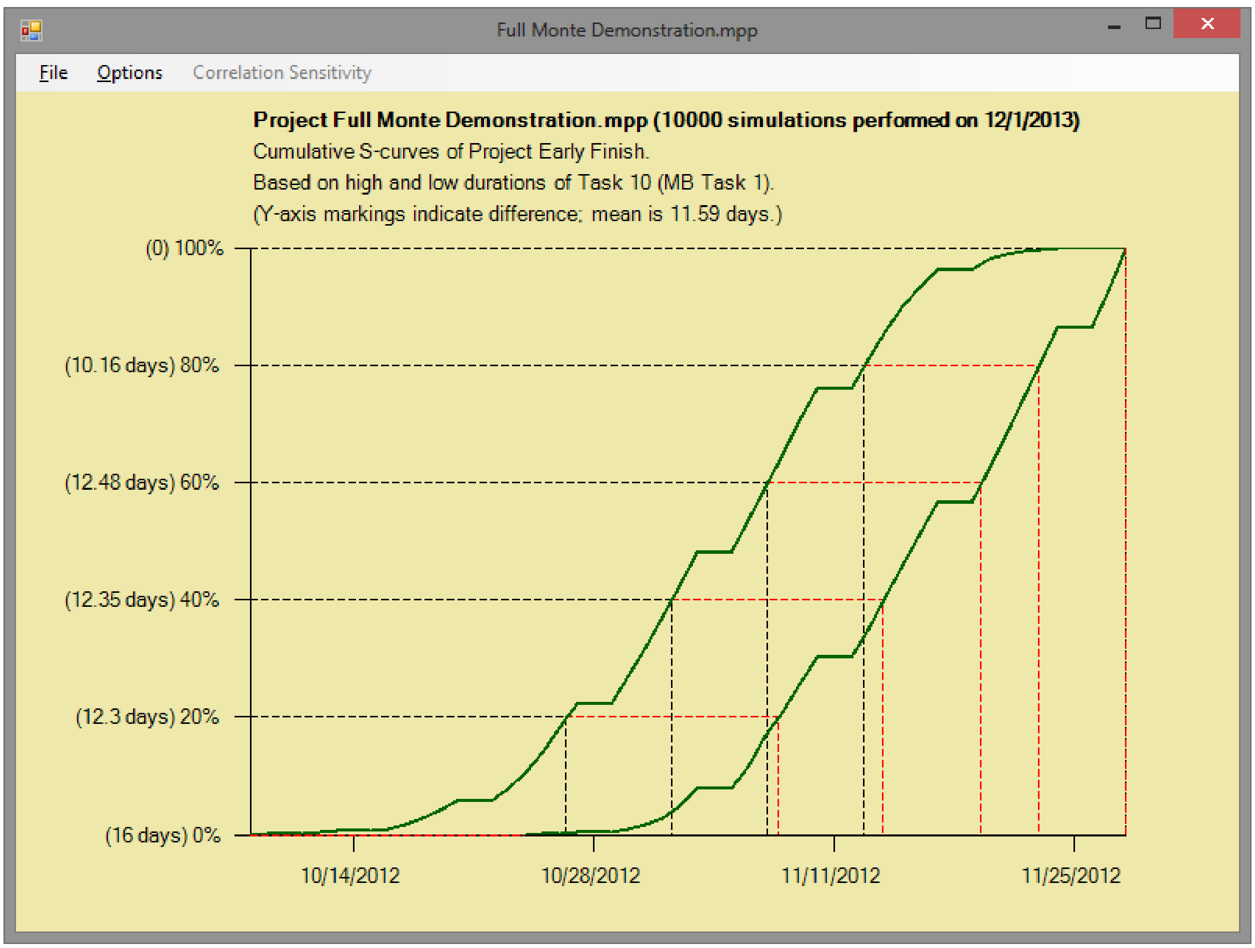
The only way to do this properly is to do two full simulations, say 10,000 trials each. It would clearly be impractical to do this for every task in the network.

Full Monte takes a two-stage approach. It uses a statistical technique to estimate the effect of each task on the end date during the main simulation. It then lists all tasks for which this was statistically significant on a chart called a tornado chart, as illustrated below.



ID	Name	Unique ID	Sens. Index, Project	Sens. Var. Contr., Project	Sens. Low, Project Completion.	Sens. High, Project Completion.
11	MB Task 2	16	0	4.27	05Nov12	03Dec12
20	Task 1 (100%)	55	7	24.69	02Nov12	28Nov12
17	Task 1 (71%)	60	18	16.43	30Oct12	19Nov12
18	Task 2 (71%)	61	49	28.94	01Nov12	16Nov12
21	Task 2 (100%)	56	50	25.42	01Nov12	16Nov12
10	MB Task 1	15	18	3.09	06Nov12	15Nov12
13	MB Task 4	19	17	3.31	07Nov12	15Nov12
12	MB Task 3	18	17	2.47	06Nov12	14Nov12

Due mainly to correlations, the effect may be overestimated. However, as long as one has done a reasonable number of iterations there will be no false negatives. Every task which can have a significant effect on the end date will be on the chart, which therefore can be used as a shortlist of tasks to investigate further. Clicking on any bar will cause Full Monte to do a full analysis – i.e. the two full simulations described above – for any task in this short list. The result is displayed as a pair of CSF curves as illustrated below.



The role of sensitivity analysis is analogous to that of the critical path in conventional deterministic scheduling. The projected end date tells you when you expect to be finished, but the critical path (and the sensitive tasks) tell you what you need to keep your eye on.

But sensitivity analysis has an extra benefit. We started out by mentioning that chance is just an expression of our ignorance. That ignorance can often be reduced, albeit at a cost. But if uncertainty about the duration of a task can have a significant effect on the end date, that cost might be worth paying.

Maybe by doing more research, or hiring an expert, we can reduce our uncertainty about that task and therefore about the end date of the project.

Risk Analysis and Progress

Risk analysis is probably most useful, and certainly most often performed, before the project starts, but there is no reason it should not be updated as the project progresses. It does raise some technical issues however.

First of all, how do we deal with in-progress tasks? If a task was meant to take 4 to 6 weeks and after 3 weeks it is 50% complete, what do we assume about the rest of the task? The best answer is to re-estimate, but that is not really practical. What we actually need is the conditional (or "partial") distribution of the total duration given that it has taken 3 weeks to get to 50%, and there is no way to answer that using the information available, for at least two reasons:

- The initial uncertainty may have been disproportionately associated with a particular part of the task; and
- Performance on the second half of the task may or may not be correlated with that on the first half.

By default, Full Monte merely pro-rates the distribution, so in the above example we would assume that the remaining work will take 2 to 3 weeks. This actually amounts to assuming a moderate positive correlation between the two halves of the task. It can be modified if desired by re-estimating the distribution for the remaining part of the task.

But the best practical solution – which is good practice with or without risk analysis – is to keep individual tasks short. That way, at any point in time most tasks are either complete or not started, so that the impact of in-progress tasks is minimized.

A knottier problem involves correlations. What does one assume if we have say 6 tasks correlated together and 3 of them are complete, so we know their actual durations? Again, we need the partial distributions of the remaining three durations, given the known values of the first three. This time we do have the necessary information, but it is hard to process.

Actually it is even harder than the above makes it sound, because there may be overlapping sets of tasks correlated with one another, so that the only valid approach is to consider the complete correlation matrix for all tasks. From this one could in principle calculate the partial multivariate distribution of the durations of the remaining tasks. Without going into detail, this involves inverting a large matrix, which in practice may well turn out to be singular. (i.e. because multiple tasks may have been correlated to the same extent to each other, while the simulated results may not exactly reflect this, it may not have an inverse.)

This is on the bleeding edge of Monte Carlo simulation, and to my knowledge no software deals with it.

Risk Analysis and Resource Constraints

I have had a number of arguments on-line about this. Clearly it is desirable to obey resource constraints in a Monte Carlo simulation, but most products do not and I know of none that do it properly.

As a practical matter, resource scheduling takes orders of magnitude longer than time analysis, so doing it thousands of times is probably not practical. But there is a more fundamental problem, namely that the assumptions of risk analysis and of traditional resource scheduling are fundamentally at odds.

Most if not all resource scheduling algorithms assume that the durations of all tasks are known precisely in advance, and make use of this fact to “optimize” the schedule. (Though actually they mostly do a poor job of this.) The point is that the duration of a task in the distant future, assumed to be known, can affect decisions on which task to schedule now.

Simply embedding a traditional resource scheduling algorithm in a Monte Carlo simulation therefore means that simulated scheduling decisions are made based on simulated information which would not be available at the time the real decision would have to be made.

I think the solution is to include a very simple algorithm for allocating resources, based only upon information that would be available at the time. This algorithm might have parameters which could be tuned, based upon Monte Carlo simulation. Look for such an algorithm in future Full Monte releases.

In Conclusion

I hope I have explained why I think schedule risk analysis is important and you will be moved I give it a try. There are a number of software products around, many with free trials available for download from the web.

You might also be moved to read more about it, and I supply a reading list below.

Further Reading

Project Risk Management
Yuri Raydugin, Wiley, 2013

(This new book covers quantitative schedule risk analysis much more thoroughly than any other book I know, devoting about 80 pages to it.)

The Failure of Risk Management
Douglas Hubbard, Wiley, 2009

(This book is not about project management specifically, but it does mention it in passing. It also has some very useful exercises to calibrate your ability to make subjective probability estimates.)

The Flaw of Averages
Sam Savage, Wiley, 2012

(This book is an entertaining read about the danger of using single-point estimates. Again, it is not specifically about project management, but it does cite it as an example of what Sam calls the "Strong Flaw of Averages.")

Modeling Activity Times by the Parkinson Distribution with a Lognormal Core: Theory and Validation.
Dan Trietsch et al, European Journal of Operational Research, 2011.
http://mba.tuck.dartmouth.edu/pss/Notes/LognormalParkinson.pdf.)

(Empirical evidence for the use for the lognormal distribution, modified to encompass Parkinson's Law.)