# FOR MORE ACCURATE SOFTWARE ESTIMATES, AVOID HIDDEN RISK BUFFERS

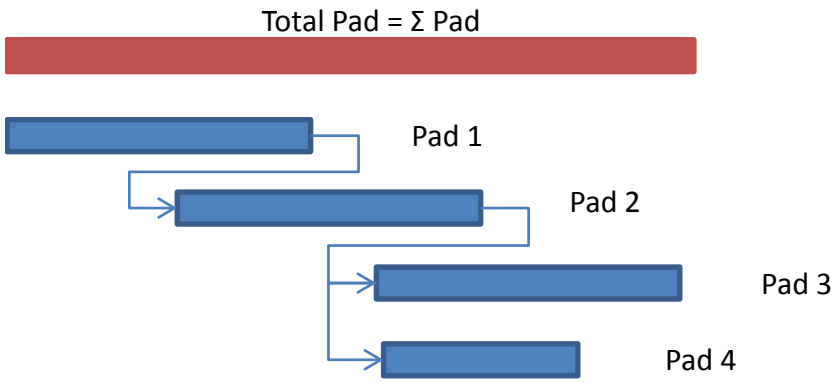
By Laura Zuber,
Senior Consultant

QSM®

A colleague of mine recently sent me a blog post explaining the difference between project *contingency* and *padding*. The blogger made the distinction that padding is what often gets added to an individual's estimate of the effort required to perform a task (in her example, a software development task) to account for project 'unknowns'. The estimator determines the most likely required effort, then pads it with a little more effort in order to arrive at an estimate to which he or she can commit. Thus, padding represents an undisclosed effort reserve (and implied schedule reserve) to buffer against potential risk. Contingency reserve, she explains, is "**an amount of money in the budget or time in the schedule seen and approved by management. It is documented. It is measured and therefore managed.**" Ms. Brockmeier is correct in promoting contingency as the better management tool. The challenge is having a method to measure and document this contingency and the known unknowns it is buffering.

## Implicit Risk Buffer

Padding is a natural result of bottoms-up, effort-based estimation techniques. Estimating low-level WBS elements creates more opportunity for padding, due to a larger number of unknowns. The estimator is consciously or unconsciously assessing the risk of each task, considering its dependencies and complexities. The cumulative effect is that a bunch of small pads rolled up to the summary task and project level equate to a large, undocumented pad that fails to trace to any particular risk scenario.




The element of human nature involved in speculation and judgment also promotes padding. Here is an example: You have identified the functionality to be included in the next product release, perhaps from a combination of release backlog and new product features. You ask your experts for an effort estimate to build each feature. You may even use a form of Delphi to garner a range of estimates. To respond, estimators must essentially ask themselves two questions: 1) To what extent is this set of features (and associated tasks) like work I have done before, and 2) How many hours did it take to produce those like features? What is being *implied* in the answers to these questions is: 1) an assessment of product size and complexity, and 2) a productivity valuation.

With this approach, however, there is no way to know what the estimator is thinking, nor what he or she has factored in to cover the unknowns. Many times the padding results not from an intentional,

conscious decision, but from unconscious experiential knowledge. It is a gut feeling that varies with each person supplying an estimate. Another cause of variability in these estimates is a less than perfect memory of past performance. In Your Money and Your Brain, Jason Zweig explains how we humans tend to be overconfident about we know.

> "Years ago, college students in Oregon were asked: Was Adonis the god of love or vegetation? Do most of the world's cacao beans come from Africa or South America? A quarter of the students were at least 98% sure that Adonis was the god of love; over a third were at least 98% certain that most chocolate originates in South America. (Try it yourself: How sure of your answers are you?) Even when they were coached about how inaccurate most people's judgments are, many students were so sure of their answers that they were willing to bet $1 that they were right. But only 31% correctly identified Adonis as the god of vegetation; just 4.8% rightly pinpointed Africa as the leading source of cacao. The same ignorance of our own ignorance haunts our financial judgments. Among American workers who say they are "very confident" that they will have enough money to live comfortably in retirement, 22% are currently saving nothing for that goal, and 39% have saved less than $50,000. Another 37% have never even estimated how much money they will need to retire comfortably. It's bad enough to be "very confident" of a cozy retirement when you are not saving for it now. It's even worse when you have no idea how much money you will need to retire comfortably—but assume you will have enough anyway. This kind of overconfidence can lead to drastic undersaving and a threadbare retirement riven with regret. That's why the old proverb, "It ain't what we don't know that gets us into trouble, it's what we know that ain't so," ain't exactly true. What really gets us into trouble is not even knowing what we don't know."

Collecting and analyzing project metrics is becoming a more common practice. Thus, many organizations have guidelines for estimating effort for a variety of product types based upon proven past performance, reducing the amount of subjectivity in estimates. Nevertheless, the fact that effort estimates in particular can be tied to peoples' performance means that some element of subjectivity is present. This is especially problematic now that most organizations are using very small teams, because the productivity number used in the estimate is traceable back to one or two people, whose performance is now in the spotlight. Estimators may underestimate to look good, or overestimate with padding.

## Explicit Risk Buffer

QSM's SLIM-Estimate tool provides an alternative approach that eliminates padding by taking a top-down view of the project. At the project level, as opposed to the task level, the vast majority of unknowns are attributed to just two factors: the product size, and team productivity. SLIM productivity, known as the Productivity Index (PI), is calculated from completed projects. All that is required is Total Effort, Total Product Size, and Total Duration, all of which are known values.

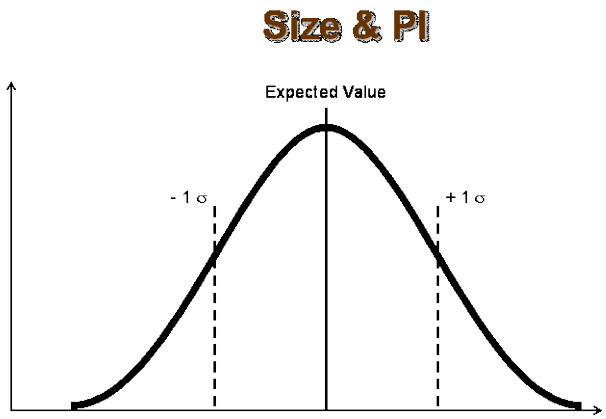
$$\textbf{Productivity} = \frac{\textbf{Size}}{\textbf{Effort}^{a} \times \textbf{Duration}^{b}}$$

With this approach, PI measures all environmental factors that affect team productivity, such as tools and methods, technical complexity of the product, management and personnel, and integration issues. Estimating the product size can be challenging, particularly early in the life cycle when little data is available. However, it keeps the focus on the problem to be solved, not what detailed tasks will be performed. The risk associated with unknowns about the product can be systematically identified and mitigated as the project progresses.

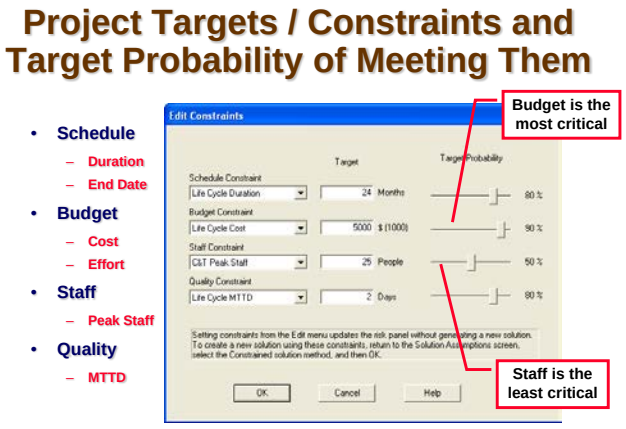
Using project productivity values measured from past projects, plus an estimate of the entire product size, SLIM calculates the total effort and duration needed to complete the project. Risk is explicitly calculated using expected values for size and productivity (along with uncertainty ranges for each input) to produce a probability distribution of effort and duration outcomes.

This probabilistic project estimate gives management the data they need to set the project contingency reserve based upon the level of risk they choose to accept. There are three opportunities to document and manage the risk:

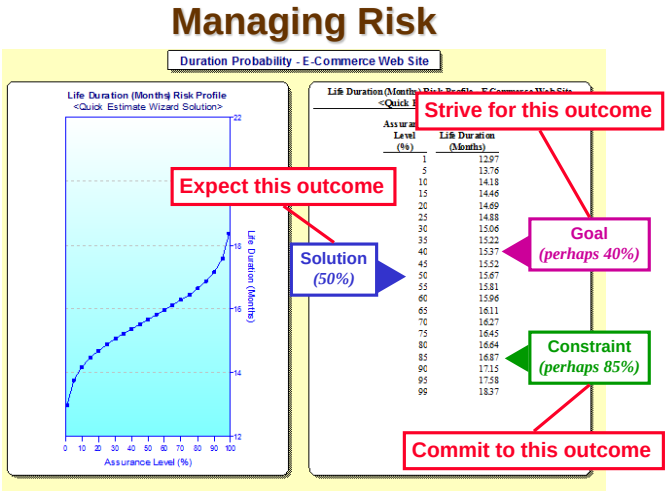
1. Specifying the uncertainty percentage for the inputs (700 function points +/- 15%; PI of 17 +/- 1.0)



2. Selecting solution assumptions that meet the project goals




3. Committing to delivery dates and costs that match risk tolerance and mitigation needs




SLIM promotes data-driven decision management by producing quantified, unbiased estimates. Subjectivity and personal opinion are minimized with the SLIM estimation process. Management knows the most likely outcome of the project (50% solution), thus can use the probability curves for time and effort to set an appropriate contingency buffer. Task level effort and duration estimates are calculated from project totals to produce a high-level project plan.

Another benefit is the ability to improve the estimation process, because variances between the original estimate and project closeout values can be traced back to the uncertainty in size and PI. There are other risk factors, of course, but these have the greatest impact. Improved sizing estimation techniques leads to more robust estimates. As more projects are completed, capturing just Five Core Metrics[1]

[1] Five Core Metrics, the Intelligence Behind Successful Software Management, Lawrence H. Putnam and Ware Myers, 2003.

produces a simple, yet meaningful database to support defensible estimates, process improvement, and performance benchmarking.

## CONCLUSION

Accuracy in software project estimation is critical to success. Use both a bottoms-up and a top-down approach to validate one against the other. Capture history for a sound basis of estimation, and use it to sanity-check estimates. The advantage of SLIM is the little amount of time and data required to explicitly express project risk, so buffering against it is a conscious management choice.

*Laura Zuber has 20 years of experience in software development consulting and training, six of which have been with QSM. She has conducted training and demonstrations for all QSM SLIM Suite Tools and assisted with QSM Support. Prior to coming to QSM, Laura managed software development projects and served as a senior software process improvement specialist at SAIC. She has performed process assessments, designed and implemented best practices, and co-lead the corporate metrics training program. Laura holds B.S. and M.S. degrees in Petroleum Engineering from Texas A&M University, where her master's thesis was on risk analysis using advanced Monte Carlo simulation techniques.*