Schedule and Cost Buffer Sizing: How to Account for the Bias Between Project Performance and Your Model

Larry Leach, Advanced Projects Institute, 5239 South Pegasus Way, Boise, Idaho 83716 USA

Keywords: buffer; contingency; management reserve; PERT; Monte Carlo; critical chain, CCPM

Abstract

Bias in project performance causes schedule and cost to overrun baseline estimates (your model). Bias is the one-sided tendency of actual schedule or cost to overrun the model. The *PMBOK® Guide* and supporting literature recommend estimating the variability for all project time and cost estimates, and sizing appropriate schedule or cost buffers (also known as contingency or management reserve) using Monte-Carlo analysis or PERT. Critical Chain Project Management (CCPM) uses a similar approach to size buffers (the square root of the sum of the squares, or SSQ method). These techniques pool the variance from individual task estimates. Statistical pooling of variance does not account for sources of bias in the estimates; i.e., systematic reasons that the estimates may be high or low. [The one exception to this can be applying Monte Carlo to the schedule. It can account for merging bias, if performed using the network.] This paper describes a number of sources of bias in performance of projects to schedule and cost estimates, and provides recommendations to size buffers that ensure your projects come in under your baseline schedule and budget.

Introduction

Projects frequently overrun the baseline schedule and budget. Despite the *PMBOK® Guide* recommendation to state the uncertainty of all estimates (PMI, 2000, p. 73), many projects use single-point schedule and cost task estimates with no estimation of variation. In reality, all project costs and activity durations are variable. In addition, project duration and cost *estimates* are uncertain … in part because of the duration and cost variability, but also due to the limited ability of the estimate to accurately predict the real variation. While schedule modeling tools such as PERT, Monte-Carlo, and Critical Chain make explicit attempts to model the variation, project performance evidence (i.e., schedule and cost overrun) indicates that they frequently miss something. This paper proposes several factors that cause schedules to systematically overrun estimates, estimates the contribution to these overruns, and proposes a method for project schedule and cost estimates to account for these causes. It also addresses how to combine the correction for bias with a buffer for normal variation in task duration and cost. The following section describes and compares the various models, and the subsequent section proposes a number of potential causes of bias in schedule and cost performance relative to estimates. The paper concludes with recommendations for sizing schedule and cost buffers to account for both variation and bias.

The Models

Projects use models (e.g., CPM) to estimate (predict) schedule and cost. Most project management literature recommends project schedule and budget estimates include specific buffers as allowances for a contingency reserve (CII, 1989; PMI, 2000), which I call a buffer. Most of these recommendations assume using the Critical Path Method (CPM). CPM generally attempts to use low-risk (pessimistic) individual task duration and cost estimates. In the discussion below, the CPM corrections assume a CPM schedule made up of low-risk task estimates summed along a path (schedule) or budget summed for the entire project (cost).

The PERT model uses optimistic, most likely, and pessimistic estimates of task duration or cost to estimate the mean and standard deviation for each task. The PERT method assumes that the difference between the optimistic and pessimistic estimates is some multiple of the standard deviation. The method then applies the square root of the sum of the squares (SSQ) of the differences to estimate standard deviation of the sum, be it the length of the critical path or the total project cost. Some texts recommend using this standard deviation and the normal distribution to estimate total duration or cost vs. probability, and to size the appropriate buffer.

Although often illustrated for cost, the same PERT model applies to schedule estimates along a path in the project; e.g., along the project critical path. The calculations start with estimating the minimum or optimistic (O), most likely (M), and pessimistic (P), time or cost for each task. Optimistic means lower (duration and cost) and pessimistic means higher cost or longer duration. I take issue with these categorizations, as in a statistical or mathematical sense there should be no attribution of motivation or emotion on these numbers. The various estimates are merely points on a statistical distribution. As points, they have equally likely probability: zero. Legitimate statement of a finite probability requires a range for the variable; e.g., "the probability that the result will be less than or equal to x is y." Or, "the probability that the result will fall between x and y is z."

Most discussions of PERT, including those in the current version of the *PMBOK® Guide* (PMI, 2000) and supporting documents (Wideman, 1992), assume six standard deviations between the extreme estimates. I will clarify below how this is one cause of bias. They recommend calculating the standard deviation, s as:

$$s = \frac{(P-O)}{6}$$

The common recommendation for the mean activity time or X_{bar} is usually stated as deriving from the beta distribution and given as:

$$X_{bar} = \frac{O+4M+P}{6}$$

Meredith and Mantel (1995) note that the above equation "could not be derived from the formula for the beta distribution without several assumptions that are not necessarily reasonable" (p. 344). The parameter assumptions for this trace back to the original PERT publication (Malcolm, Rosebloom, Clark, & Fazar, 1959), and are not addressed by many recent publications (e.g., *PMBOK® Guide*, 2000). This deficiency is well known by researchers, and alternative "more accurate" equations have been proposed (e.g., Huang, 1999). I put more accurate in quotes because, in my experience and as noted by some of these researchers, the uncertainty in actual task data usually far outweighs the variation introduced by the choice of mathematical model.

Exhibit 1 illustrates an example PERT calculation. When applying the standard CPM model, the individual task estimates (cost or duration) must use the pessimistic estimates, or P column if the project is to complete on time and within budget. So, if this were the critical path task durations, the total CPM project duration would have to be 45 days.

Exhibit 1. Example of PERT Calculation

The calculation of the total standard deviation uses the statistical method for pooling of variances. It calculates the standard deviation for the SSQ of the standard deviations for each element included in the sum.

Although not stated in most PERT discussions, the nominal values you use to communicate the schedule or cost should be the mean (X_{bar}) values. For example, you should put the X_{bar} values into your schedule for the duration estimates. The reason is that statistically, only the mean values add in a straightforward way. Thus, the schedule for the above as a chain of tasks sums to 27.5 days.

You only have about a 50% chance of completing within the nominal PERT schedule of 27.5 days; so you must add some number of standard deviations to that amount to estimate the duration for the total project. Authors recommend using two or three standard deviations, so you should add up to 7.5 days to estimate the project completion time, for a total PERT project duration (including buffer) of ~35 days.

Monte-Carlo analysis applies a more sophisticated method to implement the idea underlying PERT. Instead of fixed formulas, you estimate a distribution for each element, and instead of a defined calculation, Monte Carlo runs a simulation that randomly selects points from the individual distributions to create a distribution for the sum. Some Monte Carlo simulations allow simulating the schedule as well as the cost. The schedule simulations should account for the schedule path merging effect: an effect not important to cost calculations. Path merging means a critical path activity has additional predecessors from one or more non-critical paths through the project. Early PERT calculations did not account for this merging (Wideman, 1992), as they only looked at the critical path. I define and discuss the merging effect next.

Critical Chain Project Management (CCPM) includes buffers as essential features of the process. The CCPM buffer sizing method uses the same statistical basis as PERT, but only two estimates for the task duration: a most likely and a low-risk estimate. CCPM (Leach, 2000) does not make a specific assumption about the number of standard deviations between the two numbers. It assumes that the same number of standard deviations is acceptable for the pooled result as was inherent in the task estimate ranges. The difference between these two estimates is the uncertainty, u, in task estimate (duration or cost).

Leach (2000) describes several schedule buffers and an overall project cost buffer. He recommends using the SSQ method to size buffers, along with a minimum project buffer size of 25% of the Critical Chain, and a 10% minimum cost buffer to account for bias. Exhibit 2 shows the CCPM calculation corresponding to the PERT illustration of Exhibit 1.

Exhibit 2. Example of CCPM Calculation

Do not attach significance to the results for PERT and CCPM being so close. This is a specific (unplanned) artifact of the example, and not a general result. In general, the critical chain is not the same as the critical path, so the results can be significantly different. The CPM result will always be greater than the PERT or Monte Carlo result. The CPM schedule result with resource leveling will always be longer than the Critical Chain result.

The sum of the mean estimate plus the buffer is always less than the sum of low-risk (pessimistic) estimates for the individual tasks. Therefore, the description of *adding* a buffer can be misleading. As Exhibits 1 and 2 illustrate, compared to the sum of pessimistic or low-risk estimates (CPM), the mean probabilistic estimate with buffer is always *less*.

CCPM buffer sizing recommendations follow the statistical logic used in PERT. The sizing uses two statistical facts: (1) The variance of the sum (schedule path or total cost) is the sum of the variances of the elements; i.e., task duration or cost, and (2) The Central Limit Theorem, which states that the distribution for a collection of samples will tend toward a normal distribution, regardless of the shape of the distribution the individual samples are drawn from.

When using PERT, Monte Carlo, and CCPM, you should use the mean duration and cost estimates as the baseline task estimate. The reason for this is that the mean is the only unbiased estimate of the sum along a path or for the entire project. For these techniques, the bias corrections discussed next are relative to the sum of the mean task estimates along a path, or the total sum of the elemental costs for a project. Note (in Exhibit 1) that this sum may be substantially less than the sum of the pessimistic (low-risk) estimates used in basic CPM.

The next section illustrates a major gap between the real world of project performance and the models addresses previously, leading into the author's main hypothesis that it is bias in schedule performance to plan that causes this gap. The subsequent parts of the paper pose numerous potential causes of this bias, and the final section provides recommendations on how to account for them in your project plans.

The Real World

Real projects usually have hundreds and frequently thousands of activities, and the largest projects may have tens of thousands. Schedule and cost buffers estimated by this variance pooling method become a very small percentage of the total duration and cost. As the number of activities pooled in the variance grows, the variance of the pooled mean as a percentage of the mean reduces. You can use PERT or the CCPM SSQ method of buffer sizing to estimate this impact. Monte Carlo simulations show the same general behavior; i.e., decreasing relative total variation with increasing project size.

Exhibit 3 illustrates the relative size of a pooled variance sized buffer decreasing as the number of tasks increases. The exhibit assumes equal size tasks, each with a low-risk estimate twice the average estimate. For CCPM project or feeding buffers, the number of tasks is the number in the respective chain of tasks (path). Since cost adds for the entire project, for the cost buffer the number of tasks includes all of the tasks in the project. For four tasks, the SSQ buffer is 50% of the sum of the average estimates. For 128 tasks, the SSQ buffer decreases significantly: to only 8% of the critical chain.

Exhibit 3. SSQ Buffer Variation With Number of Tasks Illustrations in project management books, the *PMBOK® Guide*, and the CII risk management booklet (CII, 1989) illustrate using the PERT method with a few activities, sometimes as few as three, and rarely more than 10. For example, the *PMBOK® Guide* (PMI, 2000) figure 11-4 shows an example with only three tasks: Design, Build, and Test. The figure shows a low, most likely, and high estimate for each task, and then illustrates how the PERT method would require a contingency of 22% above the mean, which is higher than the sum of the most likely (median estimates) estimate to achieve 75% likelihood of success. This particular example is for cost, but the same math applies for the schedule along a path (e.g., critical path or chain). For the *PMBOK® Guide* example, the individual activity range was on the order of the minimum estimate, or ± 50% of the most likely.

If you consider a more realistic project case with 100 tasks, all with that same range for each activity as the *PMBOK® Guide* example (i.e., ± 50% of the mean), the buffer (using exactly the same method) comes out to less than 4% of the mean sum for the 100 tasks.

Construction projects fare better than many other types of projects, yet construction bids on well-defined projects typically range over ±20%. Large construction projects (e.g., Chunnel, Denver Airport, Boston Big Dig) have increasingly impressive cost and schedule overruns.

The PERT, Monte Carlo, and SSQ buffer sizing methods reduce the relative size of the buffer, as the number of tasks grows large. This implies that larger projects should be more likely to come in on budget and time; i.e., the trend of the model is toward increasing project success (closer to estimate) with increasing project size. However, real project data shows project success rate decreases with increasing project size. The famous Standish Group (Johnson, 1999) study calls it success that the average overrun (bias) on IT projects is down to 60%, from over 200% in 1994. It shows a strong (negative) correlation of project success with size, the opposite effect predicted by the PERT, Monte Carlo and SSQ methods. No projects over 10 million dollars were successful. Similar results obtain for other types of projects. Thus, predicted trends of the PERT, Monte Carlo and CCPM SSQ models do not correctly describe the trends of reality.

Exhibits 4 and 5 illustrate control charts for a series of IT projects completed in the year 2001 by an IT organization. This is real data for a relatively large IT organization in a company with ~ $2 billion/year revenue. These data are representative of the data reported by the Standish group. The numbers along the axis identify specific projects.

Control charts plot the data vs. the order of the data, and three reference lines. One reference line is the mean of the data. The other two reference lines are the upper and lower control limits representing the variation in the data. The control limits derive directly from the data also using predetermined formulas. For a process in statistical control, one point per thousand should be outside the control limits. Other indicators also indicate points outside the range of common-cause variation.

The organization that generated the results shown in Exhibits 4 and 5 underwent a transition at the beginning of 2001, adopting a formalized project delivery process. Thus, the projects were sorted by start date, and separate control limits plotted for those before the introduction of the process and after its introduction. The process follows the *PMBOK® Guide* and the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) (SEI, 1994).

Exhibit 4. Real Project Schedule Variation Illustrates Wide Range

Exhibit 5. Real Project Cost Variation Illustrates Wide Range

Examination of the control chart illustrates a key point. If you use mean estimates for tasks, and if you want projects to complete at or under the estimated duration and cost, you must add a buffer to the mean estimate. If you wish to have all projects come under the promised schedule and cost, the lower control limits must be at zero. Therefore, the average performance must be above zero by at least the variation amount. The variation amount equals one half the distance between the control limits. The amount of buffer must equal the bias plus this variation amount. The bias is the amount that the average is below zero without a buffer. Note that the two example charts show significant negative bias before adopting formal project management, and much less after the implementation.

Data for Exhibits 4 and 5 cost and schedule demonstrate two distinct groupings. The left half of the chart is a process that is not in statistical control. It has wide variation, and points outside the control limits. The mean is negative, illustrating a pattern of schedule and cost overruns. These data seem typical of the data that made up the Standish Chaos study.

The data on the right half of the charts indicates that the process has been brought into statistical control, and thus the data on the control charts is useful to predict future performance. Although difficult to discern due to the large scale necessitated by the earlier data, the control limits are still relatively large; ± 130 days (compared to an average duration for these projects of 162 days; thus about ± 90%), and ± 45% cost. The means (Xbar) are a 10-day schedule overrun, and a positive 4% cost variance.

The random component of this variation appears too large, and the bias component too small, relative to the models described in this paper. We shall return to this point after discussing the causes of bias.

Some of the projects in the later parts of the sample (right hand side of the charts) were beginning to use CCPM for schedule management, but none of the projects included a cost buffer. The narrowed control limits after process stabilization imply a needed schedule buffer of ~150 days, and a cost buffer of ~50% to ensure that most future projects complete within the estimates.

There is some confounding of the schedule data, as the projects completed in the year sorted by start date led to most of the shorter duration projects being on the right half of the control charts. The data trend agrees with the statements about the more global Standish Group data, in that there appears to be a trend of greater variation for longer duration (and higher cost) projects.

Further improvement (narrowing of the control limits) will occur as project teams become skilled in the project planning and delivery process, and as higher maturity level process elements are added. Nevertheless, a dramatic reduction in the control limits has occurred from introducing a defined, repeatable process.

As noted in the Standish group information and above, size does matter. Factors other than random fluctuations are clearly at work. The method of pooling variances is not in line with real experience. This difference is bias in project performance relative to the estimate assumptions. The discussion below poses some explanation of the observed real world data.

Whether one uses CPM or CCPM (Leach, 2000) influences how much bias adjustment may be necessary. This is because specific features of CCPM work to reduce the schedule impact of bias. The unique CCPM features that impact needed schedule buffer bias corrections are:

- Feeding buffers placed at the end of chains of activities that merge into the critical chain.
- Run rules that specify work on a project task must start when the activity is available and continue at 100% until complete.
- Assured in-project resource leveling through the definition of the critical chain.
- Across project resource leveling through project priority setting and staggering of project starts to the capacity of the constraining resource with a capacity buffer.
- Buffer management, which provides all project resources a priority tool enable focused work on project tasks.

This balance of this paper identifies 11 factors (there are probably more!) that influence project performance to schedule and cost baseline, evaluates the impact on necessary schedule and cost buffers, and provides recommendations for bias corrections.

Bias

A bias is anything that might invalidate pooling of variances of the individual tasks to size schedule or cost buffers. Since project success is judged by a single-sided (under-run) comparison to planned schedule and cost, we address only positive bias … factors that may systematically increase schedule or cost performance relative to plan. The following addresses 11 potential causes of overall under-estimate of schedule and cost.

A general rule of thumb for how many independent samples are required for statistical pooling of variance to be effective is about 25-30. With small samples, the sample mean and variance can vary widely relative to the population mean and variance. Larger samples more accurately portray the population. You need the law of large numbers (a statistical rule) working for you to successfully apply pooling of variance. The reason is that one low probability outcome can cause the mean of smaller groupings to vary by more than the pooled variance. This generalization depends on the relative variance of the independent trials, and the sensitivity of concern. Projects with less than 30 activities must use more sophisticated statistical analysis to obtain a meaningful estimate of the probability of success. The following discussion applies to projects with over 30 activities. Fewer activities might demand larger relative (%) buffers.

Omissions

Project plans often leave out activities essential for the completion of the project, such as documentation, handling interfaces with the existing infrastructure or other projects, or premium payments to ensure schedule. Project plan reviews will normally eliminate any items not necessary to deliver the project functions; but review of a plan often does not completely identify potential omissions.

Some organizations use extensive checklists, prior project plan templates, and lessons learned to help avoid omissions. These organizations will suffer less from this bias. Such practices are common in organizations that perform fixed-price contracts, and unusual in organizations that perform cost-plus contracts or internal projects. Project plan and design reviews are also more common and rigorous in the fixed-price project organizations.

Potential omissions vary by project type, environment, effectiveness of process control, and project manager and team experience. Companies that perform fixed-price projects often use extensive checklists and project plan reviews to minimize omissions. Consequently, omissions may range from ~5% to over 15% of the overall project cost estimate. The impact on schedule is more difficult to generalize, as the impact of minor omissions may be included in low-risk duration estimates, or omitted activities may not be on the critical path or critical chain.

Merging

The *PMBOK® Guide* (PMI, 2000) notes that merging project paths causes systematic increase to the expected duration of a project, as compared to performance of the individual paths. The bias occurs because the successor task at a merge point (where two or more tasks are predecessors to one task) cannot start until both of the predecessor tasks are complete. Exhibit 6 illustrates the impact for 2–10 merging paths, compared to the distribution for a single path … the critical path or chain.

Exhibits 6 and 7 calculations assumed a normal distribution to represent each path. This assumption results from applying the Central Limit Theorem to a chain of tasks. The mean of the distribution is 5 (days, weeks, months, years), and the standard deviation is 2. Recall that the central limit theorem ensures that the distribution at the end of a path of tasks will approach the normal distribution as the number of tasks in the path increases, regardless of distribution of the duration for each of the tasks in the path.

Exhibit 6. The Merging Effect Causes Late Bias

Exhibit 7. Feeding Buffers Mitigate Merge Bias

Note that for 10 merging paths, the mean increases by 60% over the nominal chain merge date. Even though the variance of this mean is reduced (because it is averaging multiple paths), late project delivery is a near certainty.

CCPM applies feeding buffers to the paths merging into the critical chain to reduce this bias. But, CCPM makes no specific allowance for the number of merging paths. In large projects, the number of merging paths may exceed the capability of the feeding buffers, combined with the sizing of the project buffer, to protect the project end date.

Exhibit 7 shows the effectiveness of the feeding buffer at reducing the increase in mean merged path completion time (or start of the successor activity on the merged path). This curve assumes feeding buffers of 50% of the path length. Smaller feeding buffers will be less effective at providing protection from the merging effect. Even with the 50% feeding buffers, the feeding buffer effectiveness is waning by time there are 10 merging paths. The bias grows to nearly 20% compared to the single path mean time of 5. The reduction in variance can compensate for some of the growth in the mean, if the project buffer is large enough to absorb the bias. Still, the bias is one third that of merging paths without feeding buffers.

The merging impact on schedule may also bias the cost estimate in two ways. First, it is often required to pay for resources engaged on the waiting paths. For example, if the truck is late to pour concrete, you may have to pay the concrete finishers who arrived at the job site to work on the concrete. Second, an increase to overall project duration also increases the cost of all of the Level-of-Effort (LOE) support to the project. The LOE impact on cost is not as great as the direct schedule delay impact because the direct schedule impact causes the cost for resources estimated within the project tasks to increase, while the LOE cost impact is due to extending the time of using the LOE resources, normally budgeted outside the project task network (or with a pseudo-task to not impact the critical chain).

Errors

Errors can only increase duration or cost. All projects have some degree of error in the product produced. Few projects explicitly include time or cost for error correction, other than planned test and rework cycles. Errors detected early in the project introduce much less bias than errors discovered late in the project, placing a premium on quality assurance processes that focus on error prevention, as compared to quality inspection oriented processes. Project error cost can easily range up to 50%. Schmidt and Kiemeie (2002) discuss the cost of poor quality (COPQ). “According to quality gurus such as Deming, Juran, etc., the percentage of COPQ in most companies is 20-40% [of total revenue].” The famous Standish Group Chaos study (Johnson, 1999), studied the cost of poor quality on IT projects, and noted outright project failure (i.e., 100% loss) exceeded 28%, and only about 15% of projects complete within 20% of estimate. Except for the highest quality organizations (i.e., Six-Sigma organizations), the cost of poor quality is at least 5%. Errors can introduce schedule bias if they impact time on the critical path or chain.

Over Confidence

Extensive research into people’s ability to estimate probability (Kahneman, Slovic, & Tversky, 1982; Morgan & Henron, 1990) demonstrates that people are over-confident in their ability to predict the range of outcomes. We noted earlier that PERT makes the assumes (an unstated assumption that is repeated over and over in derived works, along the line of urban legends and untrue “common knowledge”, such as Eskimo’s having 100 words for snow) that when people estimate extremes for their cost or schedule duration estimates, the range of these estimates represents plus or minus three standard deviations. If so, it would mean that >99% of the results fall within these extremes. Experiments consistently show that the actual results fall outside people's range in excess of 10% of the time, and often up to 50% of the time. Some of these experiments include experts in the respective fields, who show equally poor or worse performance than novices. This means that most range estimate are really only plus or minus about one to two standard deviations.

If you use PERT, you should divide by two or four rather than six to estimate the standard deviation because your estimates of the range are more likely to correspond to plus or minus one or two standard deviations than to plus or minus three standard deviations. Similar reasoning applies to fitting Monte Carlo distributions to cost and duration estimates. If you do the probability calculations at the task level, this effect is not significant on lager projects, as the random contribution to the required schedule or cost buffer is small compared to the bias elements discussed in this paper.

If you use critical path plans directly; i.e., without PERT or Monte Carlo (i.e., most projects), you are attempting to complete the project on time by completing each task on time. If, instead of exceeding task times or cost 1% of the time they are exceeded 10% to 50% of the time, you need to add an equivalent buffer; i.e., 10% to 50% of the path length or total cost estimate.

This bias does not directly affect CCPM, because no assumption is made about the number of standard deviations inherent in the estimate ranges.

Queuing

Queuing is the build up of a line of work waiting to be performed by resources. The build up can cause very long lines if the average demand for resources approaches the average supply (called the service rate in queuing theory).

Exhibit 8 illustrates the classic steady-state solution for a queue with Poisson arrivals and exponential service times. The vertical axis represents the number of tasks (length of the line of tasks) waiting to be worked on by a resource. The utilization is the average demand divided by the average service capacity of the resource. The surprising thing about queues is that the steady-state average line length (wait time) can be quite large, even with seemingly modest utilization of resource. Note that as average demand exceeds only 80% of capacity, the line begins to build rapidly. Many organizations seek to maintain 80% "billability" of resources to project work. This practice ensures that projects will have to wait for resources, and may cause project resources to multi-task, with much more serious consequences (addressed next).

Exhibit 8. Queuing Causes Late Bias at Modest Resource Loading

My own informal survey of project managers at PMI meetings around the country confirms that many CPM projects do not level resources. This means queues are very likely to form for any resource in a project. Many multiproject organizations do not use resource capacity to control the release of projects into the organization, leading to the potential for resource queues with tasks from multiple projects. In both cases, normal fluctuations in actual task duration will cause temporary queues, even if average resource loading is as low as 80%.

CCPM always levels resources within a project. This is how CCPM determines the critical chain. CCPM also seeks to explicitly avoid queues for the constraining resource in a multiproject environment (called the drum resource in CCPM) by staggering project starts using a capacity constraint buffer between scheduled demands for the drum resource on different projects. A small queue for the drum resource maintains organization efficiency by assuring that the drum is not starved for work (at the expense of project delay). Under-sizing the capacity buffer can lead to queues for the drum resource.

Although other resources in the organization are, by definition, more lightly loaded than the drum resource, random variations in task performance can cause lines to build up for tasks by these non-constraint resources. This can also lead to schedule delay. It introduces a schedule bias because there is no such thing as a negative queue; resources can't work on inputs they do not have. Queuing causes a cost bias through the LOE effect described above.

Multitasking

Multitasking is a major cause of CPM schedule bias. Many CPM projects plan on multitasking; that is, the duration controlling resource is planned to work less than 100% on the project tasks. Multitasking causes project delay for three reasons: (1) *Project tasks wait while the individual works on a task in another project, or on another task in the same project.* Exhibit 9 illustrates this effect for three project tasks. All three tasks take three times as long due to three-task multitasking. Many people report much more multitasking. (2) Task *switching efficiency loss.* Studies show this may be up to 40% efficiency loss. (3) *The network delay of multitasking.* Exhibit 10 illustrates this impact and shows how CCPM resolves it by sequencing projects. The exhibit illustrates two identical simple single-path projects, assuming a finite resource of each color. In the upper case, the resource splits effort between the two projects. In the lower case, the resources focus on one project, and then go to the next project (as illustrated by the lower case in Exhibit 10.)

Exhibit 9. Multitasking Causes All Task Durations to Extend

Not accounting for the task switching efficiency, Exhibit 10 shows that both projects complete in less time by synchronizing the start (i.e., delaying the start of the second project) to enable avoiding the task switching. This is what I am calling the network effect.

Exhibit 10. Network Effect of Multitasking Causes All Project to Extend; Sequencing Accelerates Completion of Both Projects

The schedule extension from multitasking also impacts cost, both through the task switching efficiency loss, and through the LOE effect described next.

Special-Cause Risk Events

Special-cause risk events are the realm of conventional project risk management. (e.g., Wideman, CII). Special-cause risk events may cause schedule delay or cost increase. In the strict sense, project risk assessment also seeks to take advantage of opportunities. Project plans would likely build identified opportunities into the baseline plan as they are discovered, leading to a negative cost and schedule bias. Such special-cause events should not be part of the variation within the individual task performance duration or cost ranges. They are due to identifiable discrete events outside the control of the resources performing the planned project tasks. A common mistake is to confound special-cause and common-cause risk, such as addressed by PERT, Monte Carlo, and CCPM (Leach, 2001).

The overall approach to risk assessment seeks first to mitigate such risks by planning and executing actions to reduce the probability or potential impact or a risk event. The final step in risk management is to estimate a schedule and cost buffer to absorb the impact of the collective probability of the residual risk events not identified by project assumptions or specifically excluding such items from the project scope. This is an allowance for bias.

Methods used to size the buffer include PERT and Monte-Carlo analysis. Risk events are sorted to a limited number of high impact items. The discrete risk items identified in these analyses are special-cause risks. It is impossible to determine all special-cause risks, especially those in the category of “Unknown unknowns.” High impact means that the range of the impact may be many times the baseline cost or schedule of the affected task or tasks; sometimes exceeding the time or cost estimate for the entire project. Thus PERT and Monte-Carlo analysis combining special-cause variation may not suffer the SSQ effect of much reduced size as a percent of baseline. Large projects with thousands of tasks may have only tens of items in the risk assessment.

Student Syndrome

Student syndrome is the task performance bias introduced when task performers delay starting a task until they feel task due date pressure. This usually occurs because they are busy on other tasks. Then, if everything does not go right, they overrun the estimated duration. This is devastating in CPM projects, where only one late task on the critical path makes the entire project late. It is a serious problem in many IT projects because IT projects tend to be massively parallel; that is, the development of many modules proceeds in parallel. If one module is subject to this behavior, the entire project can be late.

To some extent, properly applied PERT, Monte Carlo, and CCPM exert a positive influence, relative to CPM, reducing Student Syndrome behavior by measuring performance to mean estimates rather than to low-risk estimates. Measurement feedback can activate urgent work on the task sooner.

CCPM seeks to remove this bias by emphasizing Relay-racer or Roadrunner task performance; i.e., starting a task as soon as the inputs are available (you have the baton), working on it fully until complete, and immediately passing on the result (handing on the baton to the next racer). To the extent that behavior is not achieved, Student Syndrome can contribute to schedule bias. It should not contribute to cost bias, because it does not add to the hours spent on the task. Student Syndrome may even have a positive impact on cost, as people working extra hard to catch up frequently do not charge their overtime to the project.

Policy (e.g., Date) Driven Behavior (aka Use it or lose it)

The two words most frequently heard at project status meetings are "On schedule." If you use mean estimates for duration, you should only hear this for about half the tasks at any schedule status meeting. If you use low-risk estimates, you should hear "ahead of schedule" most of the time. (All of the children in Lake Woebegone are above average.)

Informal studies of actual completion times to scheduled completion times show that up to 80% of tasks are reported as complete right on the scheduled due date. Two possible explanations of this are that (a) people make very low-risk estimates, or (b) they believe it is acceptable to hold on to work until it is "due." If (a) were true, most projects would complete very early. Indeed, many people use the evidence of late projects to support the opposite assertion; i.e., certain kinds of people (e.g., engineers, scientists) make very aggressive (high-risk) estimates. In most organizations, culture reinforces the latter understanding and behavior. That is, management gives positive feedback for work completed on time or within budget, and negative feedback for overruns. This encourages people to try to make low-risk estimates. In addition, if one delivers substantially under the budget or estimated duration, management in many organizations exerts pressure to reduce subsequent estimates. Thus, there is a bias to achieve very little of the positive variation that may actually occur. People will turn things in on time, but not significantly early. The behavior usually means that people do extra quality checks, or may "polish the apple" until the due date arrives. In some circumstances, supervisors put aside work turned in early, until the due date is near. This supervisor behavior is another manifestation of Student Syndrome.

In cost-plus contracting arrangements, including projects within matrix type organizational structures or any type of cross charging, there frequently is a requirement to keep up billability on the part of the resource managers. Thus, they are unlikely to turn work in early and lose the compensation for the authorized hours.

The date-driven bias violates a major assumption in all of the probabilistic methods to estimate the required buffer. All methods assume that task performance will follow a probability distribution that includes actual duration or cost results below the mean as well as above. Date-driven behavior thus eliminates the left side of the statistical distributions. This causes the mean of the remaining (actual performance) distribution to move to the right … adding a bias to increase both the duration and cost of the entire chain of tasks. This bias is some percentage of the relative standard deviation of the tasks; i.e., on the order of tens of percentage points of total duration and cost, depending on the amount of variation in the individual tasks and shape of the distributions. Simulations run with a normal distribution where the standard deviation was 40% of the task mean yielded a 17% bias in the aggregate.

CCPM explicitly seeks to avoid date-driven behavior by not using task due dates or authorizing specific task budgets. To the extent this is not achieved, the existing practices require accounting for bias in the cost and schedule buffers.

Failure to Report Rework Requirements

Reports suggest that there is a frequent bias to not report known changes that require rework. Ford and Sterman (1998) suggest that this is the primary cause of the 90% syndrome, where projects are able to achieve 90% complete (reported) in one unit of time (say a year), and require another unit of time (i.e., another year) to complete the final

10%. They suggest a number of implicit policies and some human behavior attributes cause this behavior, including:

- Reluctance to report bad news, coupled with, "If I wait long enough, someone else will report a bigger slip than I have."
- Fear of the messenger being killed.
- Temporary reduction of work loads, improving apparent schedule progress.
- Reduction of paperwork associated with reporting the need for rework.
- Hiding problems permits problems from other areas to take the limelight, thus allowing time to recover.
- Maintaining adequate apparent progress reduces management intervention.

Concealment behavior can lead to a bias in schedule performance due to delaying the knowledge of rework required and overloading the resources that are responsible for the rework. This can contribute to queuing bias. The impact can be large. In the actual project and simulation analyzed by Ford and Sterman, "the concealment policy delays the last portion of the project into the characteristic pattern of the 90% syndrome, causing the project to be completed 87% (35 weeks) later than originally planned and 11 weeks (17%) later than the project management without concealment. This potential bias is very situation specific. It argues for spending whatever is necessary to ensure true scope verification.

Level of Effort (LOE)

LOE causes a potential cost bias for the project. By definition, it does not impact schedule. All projects have some amount of LOE activity. This is work that occurs at a certain rate over the duration of the project, and is not associated with specific project tasks. Examples include the operation of the project office, including the salaries of the project manager, administrators, and project-level technical personnel such as communication specialists, quality assurance, safety, and environmental protection personnel. LOE costs can range up to 20% of the project cost, but usually are more on the order of 10%. Their unique feature is that they extend over the total actual duration of the project, so any penetration into the project buffer extends the LOE cost for the project. The final sections of the paper summarize all of the potential cost and schedule bias impacts, and provide recommendations for how to account for them in your project plan.

Summary of Bias Impacts

Exhibit 11 summarizes the identified potential causes of bias, and the estimated range of impact on schedule and cost buffers. The final sections of the report describe the conclusion from considering all of the above points, how to use Exhibit 11 to size buffers, and a path for future improvement of your project delivery system.

Exhibit 11. Recommend Bias Correction Adjustments to Buffers

Back to the Real World (Conclusion)

We illustrated why the model method and trend implied by PERT, Monte Carlo, and CCPM SSQ models (i.e., smaller percent buffers for larger projects) fail to account for schedule and cost overruns, especially on larger projects. We have isolated a major potential cause of project cost and schedule overruns as failure to account for identified biases in the actual performance of projects compared to the models.

The potential bias causes identified cover a wide range. Exhibit 11 summarizes the biases and shows illustrative overall ranges. Please note:

- There is not a firm technical basis for some of the estimates.
- The bias range is different for every organization, depending on the organization behavior.
- There is no theoretical method to sum the various biases.

It is very unlikely that one could sell project customers on buffers as large as 100%. I have heard senior managers criticize project managers proposing a buffer of only 25% stating, "If you can't estimate more accurately than that, we should not be doing this project." If you bid projects competitively, you would never win a contract with buffer estimates of 100% or more.

Fortunately, most organizations that do projects do not do just one project. They do many. Their project planning and delivery process may be formal or informal, it may be good or bad, and it may be in statistical control or not. Organizations can (should!) manage that process using the tools of process management. The most important tool is the Control Chart, introduced at the beginning of this paper. The Control Chart is the "Voice of the Process," in this case, your project delivery process.

Exhibit 12 illustrates a control chart for an effective project delivery process; i.e., one that delivers on or under time and cost all the time. This generic chart applies to both cost and schedule. For that case, the required buffer is the sum of (-) X_{bar} and the half-width of the control limits. Adding a buffer effectively moves the chart to the position indicated on the right side of Exhibit 12, meaning very few projects fail to come in early and under the cost estimate.

Exhibit 12. Illustration of a Control Chart for an Effective Project Delivery Process

The earlier discussion of the control charts presented in Exhibits 4 and 5 identified that the apparent random contribution to the variation was too large, and the bias contribution too small, compared to the theory presented herein. We can now unravel the explanation in two parts. First, the random variation in the individual tasks is much larger that normally asserted. Task estimates based on historical experience within a multitasking environment are a likely cause of this effect. That is, estimates are made to conform to experience. The variation introduced by multi-tasking can be several hundred percent of schedule time, not the tens of percent usually assumed for standard deviation on individual tasks. Other uncontrolled bias factors randomly turned on and off for projects add to the observed overall variation. This is how statistically summed actual variance can be as large as 50-90% after the process has been brought into control, as demonstrated by the right side of the control charts.

Second, the experientially determined bias on the right side of the charts appears too small because, contrary to assertions, bias was included in the initial estimates. Instead of including a schedule buffer at the end of a chain of tasks, or adding a cost buffer to the mean cost estimate, buffer was distributed amongst all the tasks … hidden. (This is the most common practice.) The author confirmed this with a separate analysis of individual project-like tasks that used the same estimating process. The included bias was about 30%. That is, you must cut individual task estimates to 70% of the original estimate to obtain the mean duration and cost for the tasks. Initial projects using CCPM cut project estimates to 60% and confirmed this … i.e., the projects completed well within the buffer.

Note that control charts make no assumptions about the variance or cause of variance: they are the purely empirical voice of the process. Initially, the variation within individual tasks can be so large as to mask the bias contributions, as in the illustrated example. As process improvement (including the elimination of bad multitasking) continue to reduce the variation of the individual tasks, and as estimates move toward mean estimates, the bias contribution will emerge.

With buffers, as illustrated by the right side of Exhibit 12, projects will, on average, underrun by the bias amount and only rare projects will overrun the cost and schedule estimates. Process improvement should work to reduce both variation and bias over the long term. Variation reduction will focus on the work processes within the tasks, while bias reduction will focus on improving the project estimation and delivery process.

Recommendations for Schedule Buffers

Size project schedule buffers as follows:

Project Schedule Buffer = Variation Buffer + Bias Buffer

For basic CPM, the Variation Buffer is zero. For the other methods, sum the variation component along the critical path or critical chain. For PERT, world data on overconfidence suggests using a divisor of two to four instead of six for the variance range estimate. For PERT and Monte Carlo, consistency suggests setting the variation component of the buffer to two s.

Until you are able to use the control chart in Exhibit 12, you must assess Exhibit 11 to determine which of the bias elements apply to your projects, and estimate an appropriate amount for the bias component. For example, for projects with most of the tasks on the critical path or chain, the merging impact may be insignificant. For IT projects with a hundred parallel modules, the merging impact is potentially huge. Or, if you have an excellent quality process, know your error rates, and built correction into your plan, the error correction may be negligible. If you do not have a formal quality process and data on the actual error rate, and you have not built rework into your plan, your error correction bias will be very large.

For organizations new to CCPM, a total schedule buffer equal to 50% of the chain length has proven successful for many projects. This applies to both feeding and project buffers; although it is the author's experience that this method leads to frequently penetrating the feeding buffers located early in the project plan by over 100%. The project buffer should never be less than 25% of the total task duration along the critical chain.

After you have completed 25 or more projects, you should create and use the control chart to determine the necessary buffers. You can also use the control chart to identify opportunities to reduce the variance of project schedule performance. For example, you may need to improve task identification, task estimating, task performance processes, quality processes, or risk management to reduce the bias or the variance of schedule performance vs. estimate. As you narrow the variation, you reduce the necessary schedule buffer size.

Recommendation for Cost Buffer(s)

Your point of view determines your approach to sizing cost buffers. For example, fixed-price contract projects demand a different approach than internal developments. In general, size the cost buffer as follows:

Project Cost Buffer = Variation Buffer + Bias Buffer

You should size the variation component of the buffer using the same methods described above for schedule buffer, but summed for all tasks in the project.

Exhibit 11 indicates less potential cost bias than schedule bias. Management pressure frequently urges minimizing cost buffers. If you are bidding projects competitively, you must minimize cost buffers to win the job while avoiding too small a buffer that would cause you to lose money on the job. You are between a rock and a hard spot. These factors suggest an alternative approach to cost buffer sizing; i.e., balancing the risk of losing the project revenue and profit vs. the potential financial loss from too small a cost buffer.

The construction industry, which is primarily a fixed-price project environment, commonly uses prior experience or Monte-Carlo analysis to size cost buffers. The Monte-Carlo analysis should account for common-cause variation, although sometimes special-cause variation is erroneously confounded in Monte-Carlo analysis. It may account for the merging effect, if performed on the schedule in addition to the cost estimate. Some methods separately perform the Monte-Carlo analysis on schedule and cost.

Based on the above, a minimum cost buffer on the order of 25% appears justified, if not too small. The total cost buffer should never be less than 10% of the mean cost estimate.

References

CII (Construction Industry Institute). (1989). *Management of Project Risks and Uncertainties*. CII Publication 6-8. Austin, Texas: University of Texas at Austin.

Ford, David N., & Sterman, John D. (1998). *Overcoming the 90% Syndrome: Iteration Management in concurrent Development Projects*. Downloaded from MIT WEB site January 2002. Cambridge: MIT

Huang, Yao-Chin. (1999). *The Alternatives of Estimating PERT Activity Times and Standard Deviations*. Downloaded on 3/02/02 from http://www.scis.nova.edu/~yaochin/621-p.htm.

Johnson, Jim. (1999, December). *CHAOS into Success*. *Software Magazine*.

Kahneman, Daniel, Slovic, Paul, & Tversky, Amos. (1982). *Judgment under Uncertainty: Heuristics and bias*. New York: Cambridge University Press.

Leach, Lawrence P. (2000). *Critical Chain Project Management*. Boston: Artech House .

Leach, Lawrence P. (2001, February). Putting Quality in Project Risk Management. *PM Network*, p. 53.

Malcolm, Donald G., Rosebloom, John H., Clark, Charles E., & Fazar, Willard. (September 1959). Application of a Technique for Research and Development Program Evaluation. *Operations Research*, p. 646.

Meredith, Jack R., & Mantel, Samuel J. (1995). *Project Management: A Managerial Approach*, *Third Edition*. New York: John Wiley.

Morgan, M. Granger, & Henron, Max. (1990). *Uncertainty, A guide to Dealing with Uncertainty in Qualitative Risk and Policy Analysis*. Cambridge University Press: New York.

Project Management Institute (PMI). (2000). *A Guide to the Project Management Body of Knowledge.* Upper Darby, PA: PMI.

Schmidt, Stephen R., & Kiemeie, Mark J. (2002). *Don't Let TQM Drain You Dry Without Any ROI*. Retrieved December 10, 2002, from http://www.airacad.com/tqmroi.htm

Software Engineering Institute (SEI). (1994). *The Capability Maturity Model: Guidelines for Improving the Software Process.* Reading, MA: Addison-Wesley.

Wideman, R. Max. (1992). *Project and Program Risk Management, A Guide to Managing Project Risks and Opportunities*, The PMBOK Handbook Series-Volume No. 6. Upper Darby, PA: PMI.

Larry Leach, PMP, is the president of the Advanced Projects Institute (API), a management consulting firm. He developed and operated the Project Management Office for American National Insurance Company (ANICO), in Galveston Texas, performing large IT projects. He worked at the vice president level in several Fortune 500 companies, where he managed programs of large and small projects of many types. He has Masters' Degrees in both business management from the University of Idaho, and in mechanical engineering from the University of Connecticut. He is a member of the Project Management Institute (PMI), and has published many papers and books and presented seminars on related topics.

http://www.advanced-projects.com/

Exhibit 1: Example PERT Calculation

Task	O	M	P	Xbar	s	s^2
1	2	5	10	5.33	1.33	1.78
2	3	5	12	5.83	1.50	2.25
3	4	6	8	6.00	0.67	0.44
4	7	10	15	10.33	1.33	1.78
Totals		**CPM**	**45**	27.50	2.50	6.25
			3*s	7.50		
			PERT	**35.00**		

Exhibit 2: Example CCPM Calculation

Task	M	P	u	u^2
1	5	10	5	25
2	5	12	7	49
3	6	8	2	4
4	10	15	5	25
	26			103
SSQ	10			
CCPM	**36**			

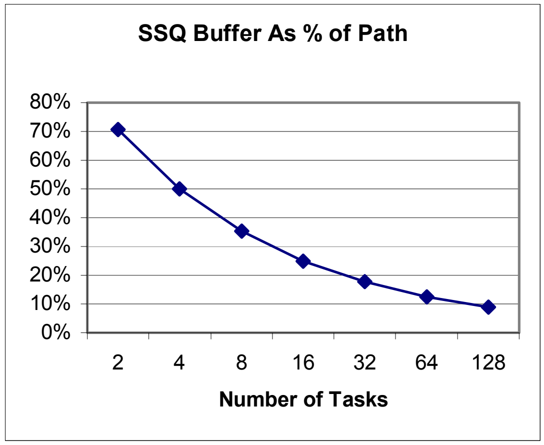

Exhibit 3: SSQ Buffer Variation with Number of Tasks

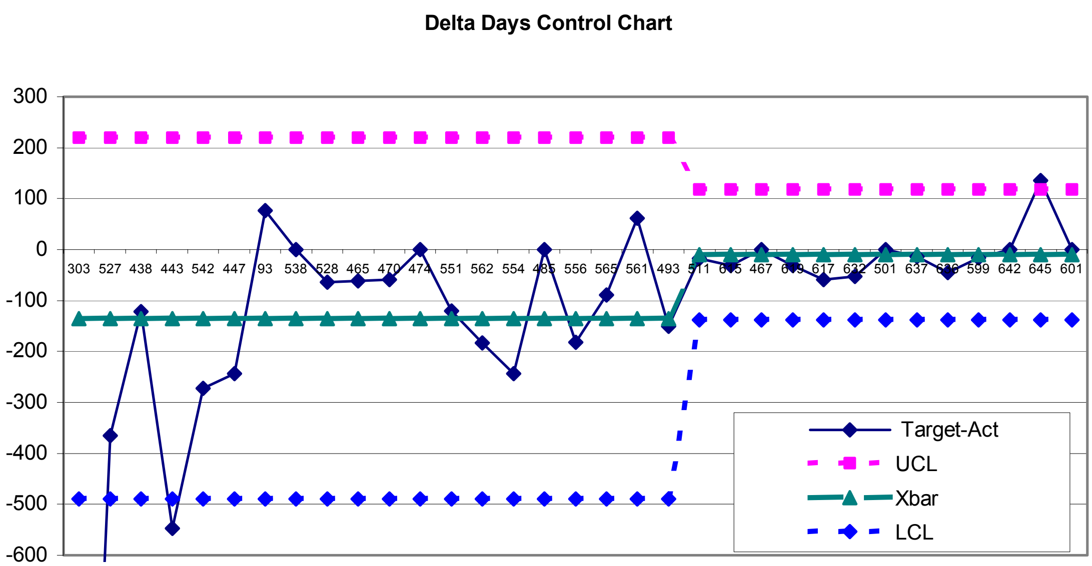

Exhibit 4: Real Project Schedule Variation Illustrates Wide Range

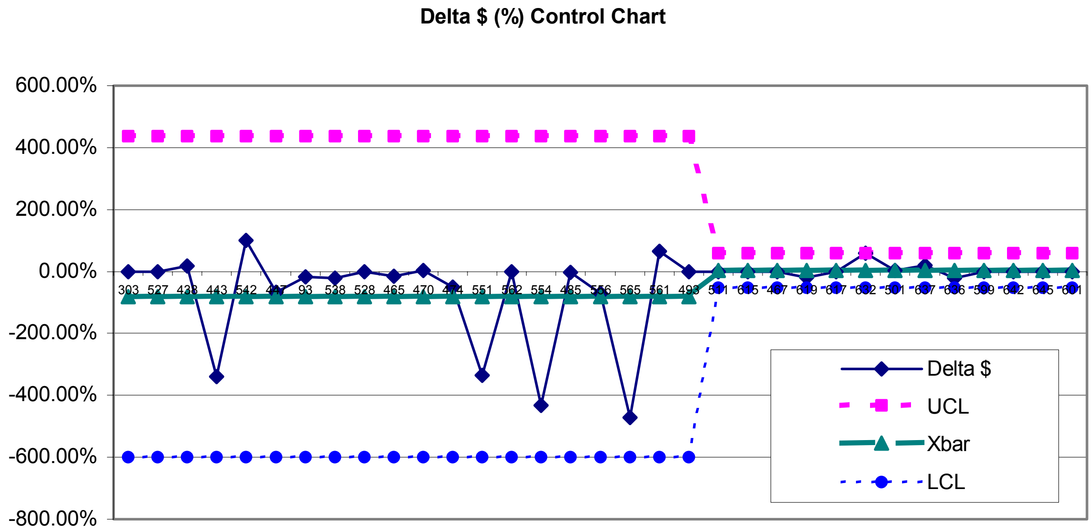

Exhibit 5: Real Project Cost Variation Illustrates Wide Range

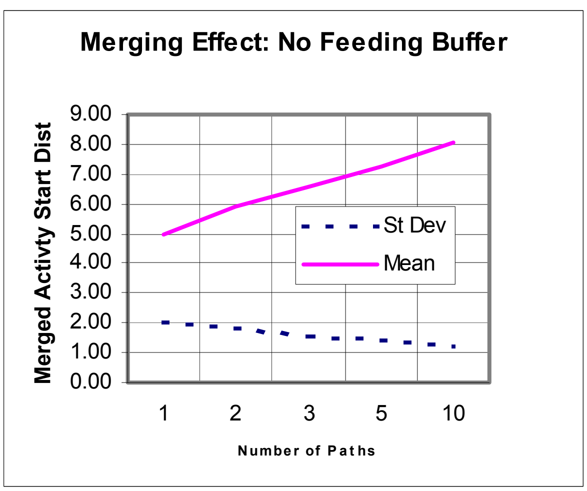

Exhibit 6: The Merging Effect Causes Late Bias

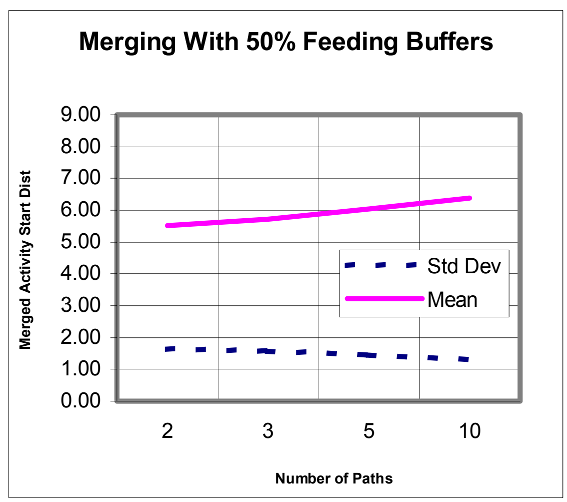

Exhibit 7: Feeding Buffers Mitigate Merge Bias

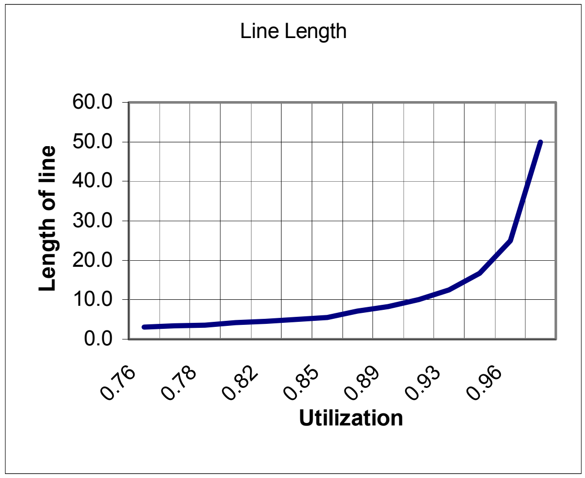

Exhibit 8: Queuing Causes Late Bias at Modest Resource Loading

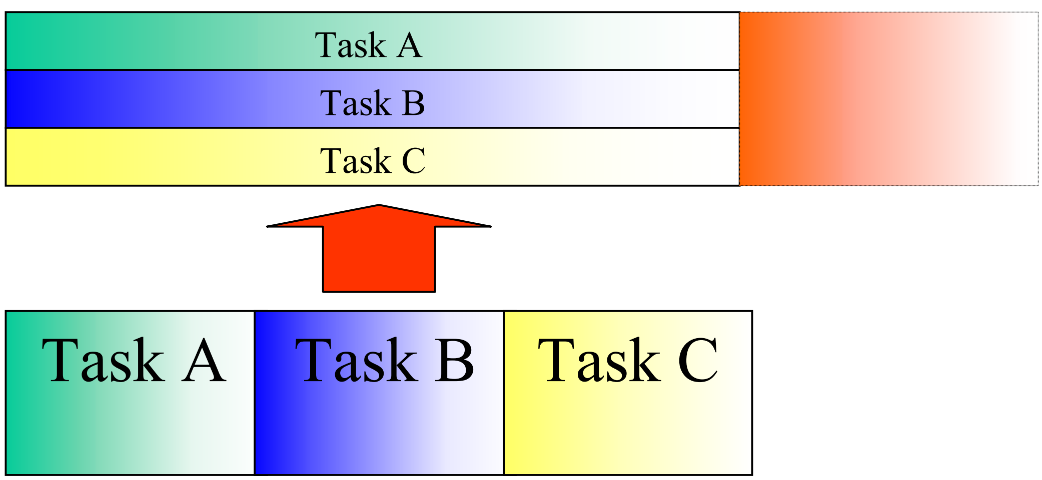

Exhibit 9: Multi-tasking Causes All Task Durations to Extend

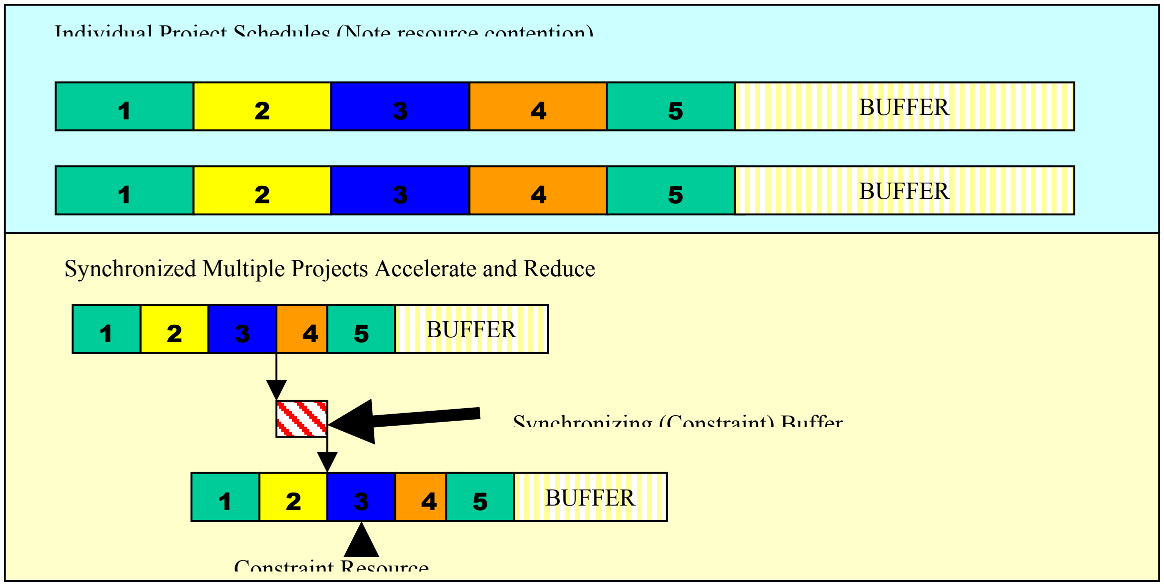

Exhibit 10: Network Effect of Multi-Tasking Causes All Projects to Extend; Sequencing Accelerates Completion of Both Projects

Exhibit 11: Recommended Bias Correction Adjustments to Buffers

Bias Factor	Schedule Impact	Cost Impact
Omissions	Some, not to exceed cost impact	5%-10%
Merging	50%+ for CPM, less for Monte-Carlo that calculates schedule < 20% CCPM	None to small (controllable)
Errors	5%-25%	5%-50%
Over confidence	10%-50% CPM More for PERT, Monte-Carlo None CCPM	10%-50% CPM PERT, Monte-Carlo ~30% None CCPM
Queuing	CPM: 100% or more if no resource leveling in or across projects CCPM: Nominal (Resource leveling & capacity buffer)	No direct impact (See LOE)
Multitasking	CPM: Several hundred % or more CCPM: Small (feeding buffers)	Up to 40% (efficiency loss), plus LOE impact
Special Cause Variation	0%-30%	0%-30%
Student Syndrome (Starting late)	CPM: 10%-20% Less for PERT and Monte-Carlo if mean estimates used in baseline CCPM: Small (Buffer management)	None to positive
Date Driven Behavior (Not reporting early completion)	CPM: 10%-20% CCPM: Small (Run rules)	~5%
Failure to Report Rework	0%-20%	Covered by errors
LOE	None	LOE rate times schedule delay
Total CPM	> 100%	10% -50% (Cost Buffer)
Total CPM (PERT, Monte –Carlo)	> 50%	10% -50% (Cost Buffer)
Total CCPM	10%-25% (Project Buffer) More with many parallel tasks	10%-25% (Cost Buffer)

Exhibit 12: Illustration of Control Chart for Effective Project Delivery Process