

Critical Chain Scheduling and Buffer Management . . .

Getting Out From Between Parkinson's Rock and Murphy's Hard Place

This article was originally published in the Project Management Institute's April, 1999 issue of PM Network by [Francis S. "Frank" Patrick](#) of [Focused Performance](#)

It can be downloaded in Adobe Acrobat (pdf) format at [CCBM_PM.pdf](#).

"Work expands to fill (and often exceed) the time allowed." -- Parkinson's Law

"Whatever can go wrong, will." -- Murphy's Law

Uncertainty is why we need project management. How we manage for uncertainty is at the core of improvement of project performance -- getting projects done both faster and with better reliability of the promised deliverable dates.

The approach to project management known as "Critical Chain Scheduling and Buffer Management" provides mechanisms to allow a "whole system" view of projects. It identifies and *protects* what's critical from inevitable uncertainty, and as a result, avoids major impact of Parkinson's Law at the task level while accounting for Murphy's Law at the project level.

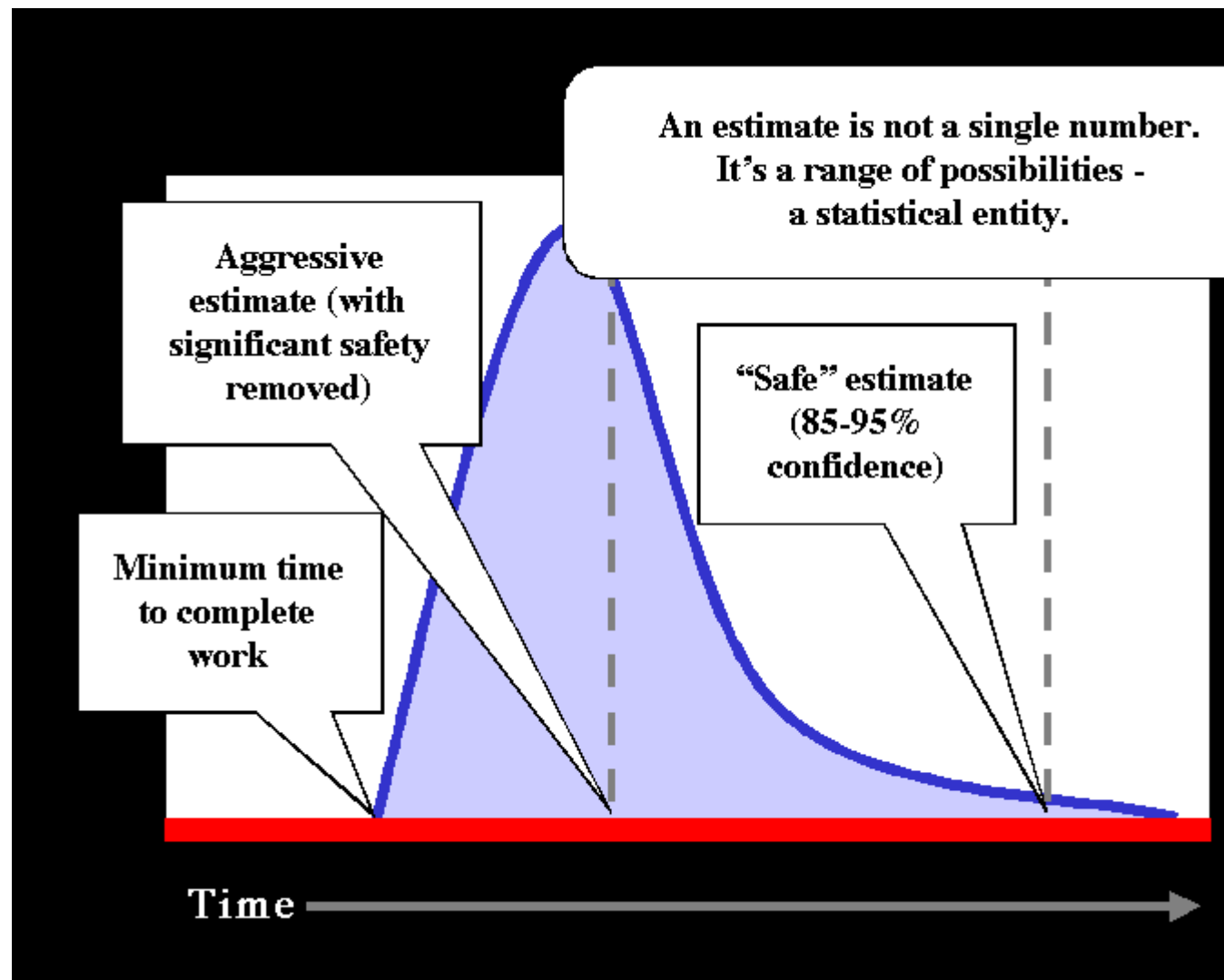
Project managers and teams need to shift their attention from assuring the achievement of task estimates and intermediate milestones to assuring the only date that matters -- the final promised due date. Safety that is typically built into tasks to cover Murphy's Law is inefficient, leading to longer than necessary (or acceptable) schedules, and apparently ineffective, given the impact of Parkinson's Law from which many projects suffer.

THE MAJOR PROBLEM AND CHALLENGE OF PROJECTS

Project management must reconcile two conflicting aspects of projects -- the increasingly important need for speed in project delivery and the equally important need for reliability in delivering the project as promised. Project management must deal with uncertainty in an attempt to deliver project outcomes with certainty. One way of thinking about how to deal with this conflict is to develop strategies to avoid expansion of project lead-time (Parkinson's Law) while protecting against Murphy's Law.

The way we manage for uncertainty in projects is at the core of improvement of project performance, defined as getting projects done both faster and with better reliability of the promised final project due date. In most projects managed with commonly accepted practices, this uncertainty is dealt with by focusing on delivery of tasks with the seemingly reasonable belief that if individual tasks come in on time, the project will as well.

Developed through the application of the Theory of Constraints to the subject of projects, "Critical Chain Scheduling" suggests the shifting of focus from assuring the achievement of task estimates and intermediate milestones to assuring the only date that matters--the final promised due date of a project. As a matter of fact, the scheduling mechanisms provided by Critical Chain Scheduling require the elimination of task due dates from project plans. One benefit is that it allows those who use it to avoid the significant impact of "Parkinson's Law;" i.e., work expanding to fill the time allowed. Take away the idea of time allowed, and you've got half the battle won. But how to do that is the question that requires us to look at some current common project practices and how they lead to "Parkinson's Law."

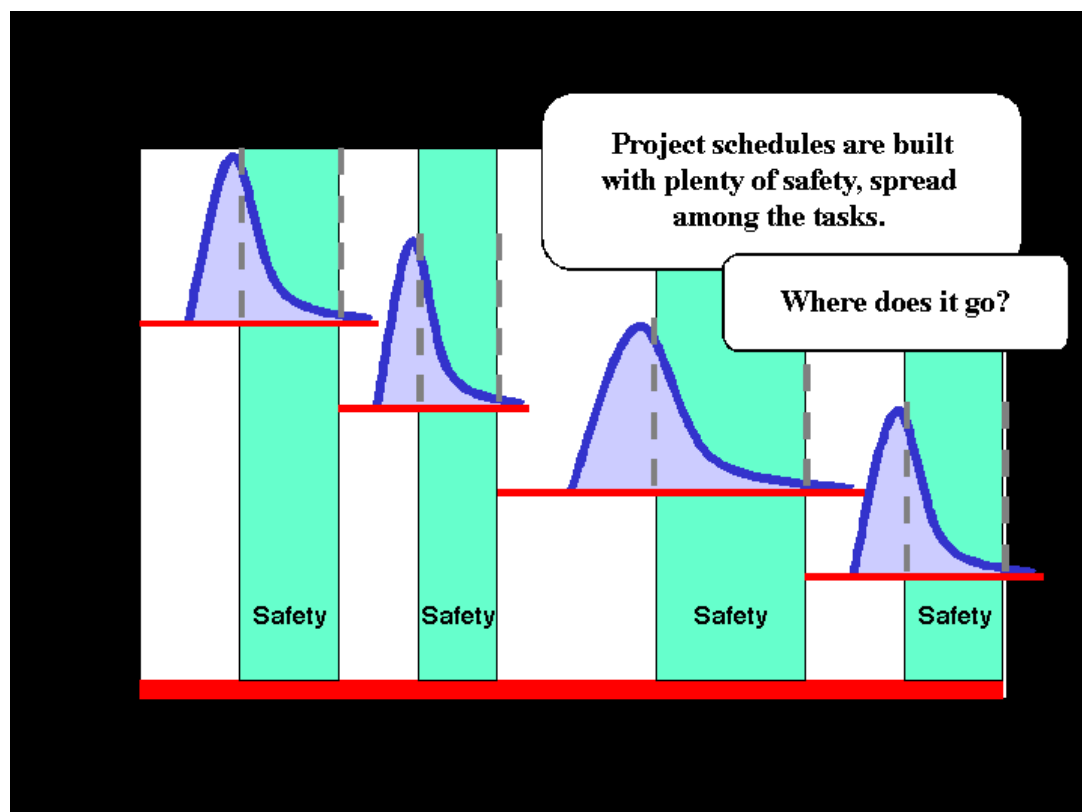


People usually derive schedules and their component deadlines from estimates of duration required by the various tasks that comprise the project. (How long will it take?) In many cases, project resources know that they will be held accountable for delivering against their estimate, and equally, that the organization needs to be able to count on their promise. Therefore, it is prudent that they include not only the amount of focused effort/time they expect the work to take, but also time for "safety" to protect their promise. This safety must deal with the uncertainty involved in the work (Murphy's Law), the impact of distractions and interruptions they live with in their organization, and, in many cases, the effect of dealing more than one such project at a time. (Have you ever been "half-a-

headcount" on more than one project? If so, the promise associated with a task on one project can be significantly impacted by time spent on the others.)

When looked at as a whole, these estimates are not really a single number, but rather they are statistical entities, reflecting the probability of task completion in a certain amount of time. An aggressive estimate, reflecting only the amount of work required might have a 50% level of confidence, while a longer realistic estimate, one against which the resource is comfortable committing to, might be closer to an 85-95% range of confidence.

So task estimates have plenty of safety in them, above and beyond the actual expected time to do the work. Often this safety is the larger part of the estimate, doubling or tripling the amount of time the work would require if done in a vacuum.

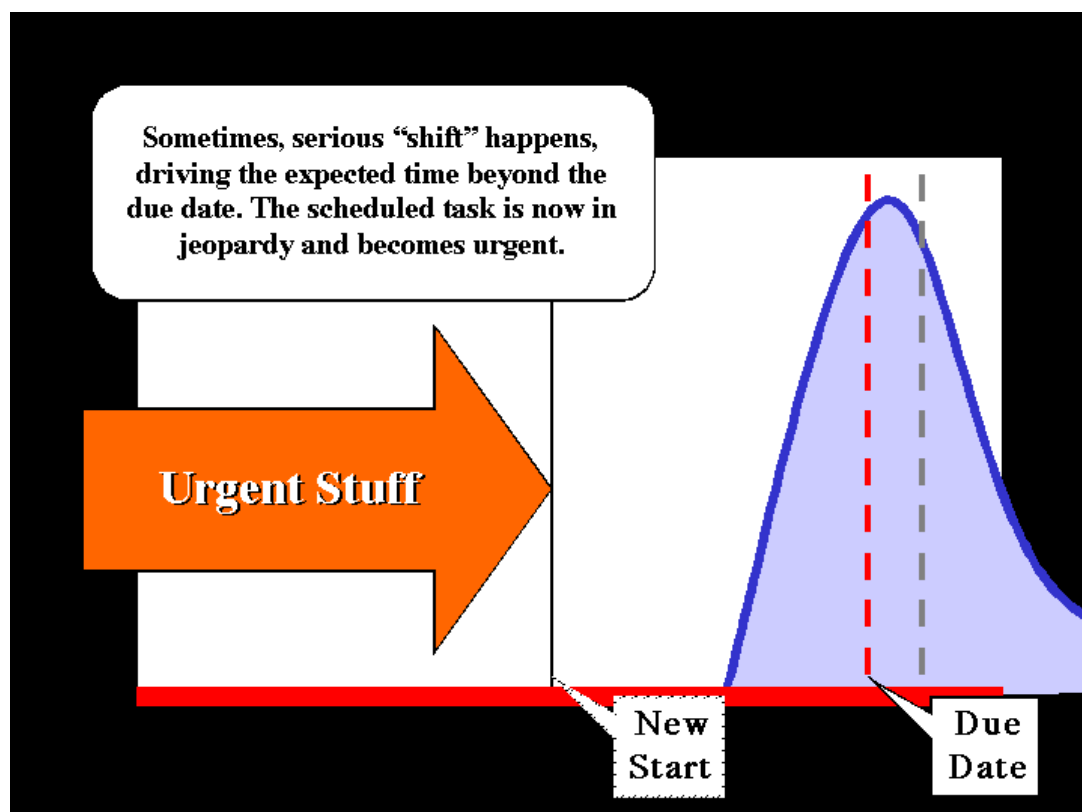


What happens to this safety? Why is it so hard to meet task deadlines and project promises? In some occasional cases, it may simply be an issue of excessive problems or erroneous assumptions overwhelming the safety, but the difficulty of bringing in projects on time is so common that there must be something else happening in the system contributing to the effect. Perhaps it's in the way the safety is used.

In most projects, estimates are turned into a project schedule -- a list of dependent tasks with associated start-dates and due-dates. People plan their work around these dates and focus on delivering their deliverables by these dates. ("Hey - What are you bothering me today? It's not due for another two weeks!") They also try to plan other work so they are free to work on the project task at the start date.

The problem comes in when the scheduled time arrives. It often happens that there is other "urgent stuff" on one's desk when the task shows up in the in-box. And in any event, we have until the promised date to finish the work, which at this point looks like a long way off due to the safety included in the estimate. We are comfortable putting off or "pacing" the work in favor of other stuff because the due date is out there.

The "urgent stuff" takes precedence until we see the due date sneaking up on us, or, as the following graphic shows, the due date is within even the aggressive expected duration of the work itself. Sometimes it sneaks up quietly enough (drowned out by the louder squeaking wheels) that when we look, we realize that it has now become urgent and gets our attention. (After all, we tell ourselves we work better under pressure anyhow, right?)



So now the originally scheduled project task is hot. If our office has a door, we close it. We let voice mail pick up our calls. We work at home to get the job done without distractions. The only problem is -- problems.

The safety that we included was not only for the non-project distractions, but also for the unknowns (the "Murphy") associated with the task itself. We can't know what problems will crop up until we start the work. And we've started the work later than planned, after eating up most, if not all, of our safety attending to other important work. There isn't time left to recover from the problems in time to meet the due date, at least without heroics, burnout, or loss of quality.

So task deadlines are hard to meet...and cascade through the project, putting the promise of the final delivery into jeopardy, which creates new "urgent stuff" which impacts other projects...and so on and so forth.

Even if, by some miracle, you do finish a task early, since the next task is keying off your original deadline as a start date for their task, will the required resource be available to pick it up? Or will they feel an urgency to pick it up, since now they have not only their safety, but also your early delivery to protect their due date? I think not. So the project is pretty well doomed to meeting the final target date at best, but in all likelihood either missing it, or just making it with burnout heroics or compromised quality.

. . . Parkinson's Law strikes!

This all occurs due to the combination of task due dates and realistic, prudent, "safe" estimates. We protect our project due dates by protecting task due dates with safety. Then, from the point of view of the project, we waste that safety due to the comfort it provides, and put the project promise in jeopardy.

If there were a way of managing projects without task due dates and the undesirable behaviors they instigate, it would have to deal with several non-trivial challenges:

- How can we systematically protect the promise date of an entire project from Murphy and uncertainty without nailing all the tasks to deadlines on a calendar, which brings Parkinson and wasted safety time into the picture?
- How can we systematically take advantage of early task finishes when they can help us to accelerate the project and maybe allow us to finish it early, freeing up the resources to address other projects?
- How can we manage the execution of a project -- how do we know what shape our project is in once it gets started, if we don't have due dates to track?

One solution to these challenges is found in the approach to project management known as Critical Chain Scheduling and Buffer Management.

CHALLENGE 1 -- ACHIEVING BOTH SPEED AND RELIABILITY

How can we systematically protect the promise date of an entire project from Murphy and uncertainty without nailing all the tasks to deadlines on a calendar, which brings Parkinson and wasted safety time into the picture?

Three things can help to avoid Parkinson's Law.

- Build the schedule with target durations that are too tight to allow/encourage diversion of attention.
- Get rid of task due dates.
- Charge management with the responsibility to protect project resources from interruptions rather than getting in their way with unnecessary distractions.

As previously mentioned, estimates typically include not only the amount of focused effort and time they expect the work to take, but also "safety" to deal with:

- The uncertainty involved in the work itself (Murphy's Law).
- The impact of distractions and interruptions they live with in their organization/environment, and, in many cases.
- The effect of dealing more than one such project at a time.

The Critical Chain methodology requires that the schedule be built with only the time to do the work without any safety. This is the time we expect the work to take if allowed to focus a full sustainable level of effort on it and if there are no significant problems. We usually describe this estimate in terms of having a 50% confidence level. (Obviously, a management paradigm shift comes into play here, because while resources are expected to strive for these "target durations," in no way can/should they be considered commitments. Otherwise, performance measurement pressures will result in building safety back in, re-expanding the estimates.)

This now leads directly to and supports the second requirement for repealing Parkinson's Law -- the elimination of due dates. There's an almost Zen-like statement associated with project tasks that suggests that no matter what any estimate says, "The work will take as long as the work takes." If we're building a schedule on the basis of aggressive, 50% confidence durations, we can't expect people to meet them all the time, and therefore there is no way we can think in terms of due dates.

CHALLENGE 2 -- EARLY FINISHES ARE TIED INTO ACHIEVING SPEED

The first two challenges cross paths at this point. The preceding discussion begs the question "Without dates, how do we know when particular resources need to be available?" This is closely related to our second challenge, "How can we systematically take advantage of early task finishes when they can help us to accelerate the project and maybe allow us to finish it early, freeing up the resources to address other projects?" Early finishes are simply a special case of not having predictable dates to tie to our activities.

In the Critical Chain world, there are two kinds of resources; resources that perform critical tasks and resources that perform non-critical tasks. The ones we really have to worry about in this context are the critical chain tasks, since they most directly determine how long the project will take. We want to make sure that critical chain resources are available when the preceding task is done, without relying on fixed due dates.

There are two simple steps required to accomplish this. Step one: Ask the resources how much of an advance warning they need to finish up their other work and shift to interruptible work so that when the preceding project task is complete, they can drop what they're doing and pick up their critical task. Step two: Require resources to provide regular, periodic updates of their current estimate of the time to complete their current task. When the estimate to complete task A matches the advance warning needed by the resource on task B,

let the B resource know the work is on its way and that it should get ready to pick it up.

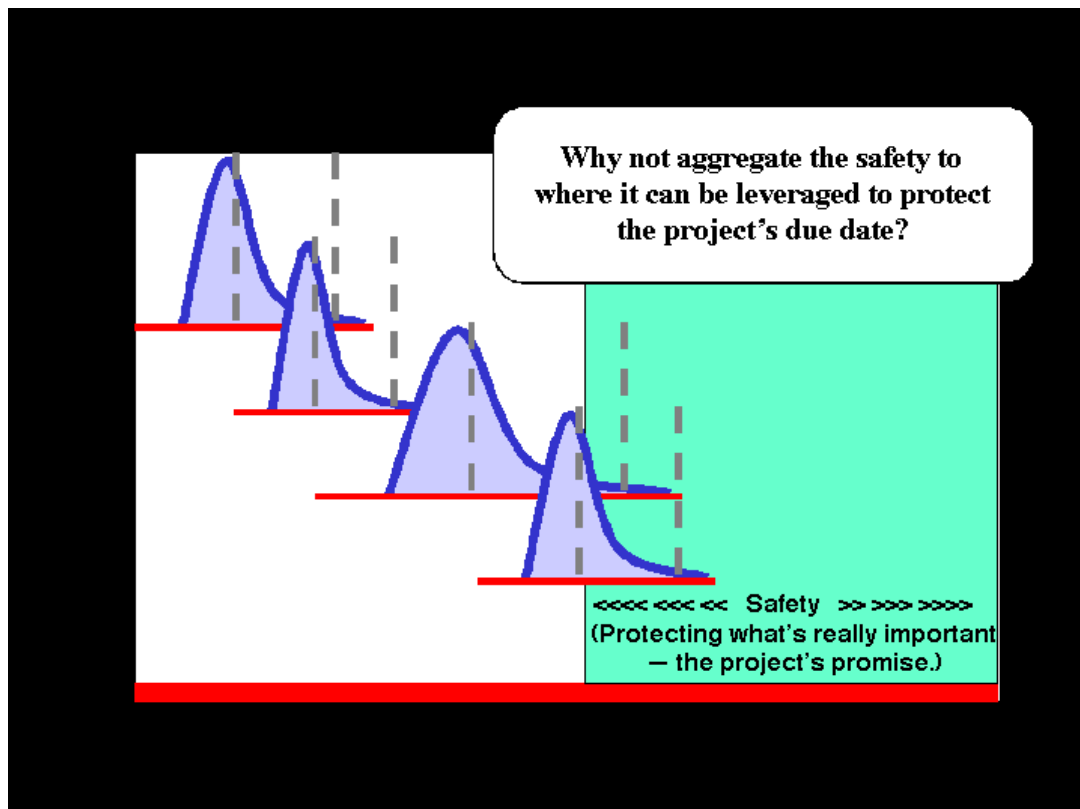
Compared to traditional project management, this is a bit of a shift away from focusing on "what we've done" via reporting percent of work complete to focusing on what counts to assess and address project status--how much time is left to accomplish unfinished tasks.

This process puts us into a position such that we're no longer nailed to the calendar through due-dates, we can move up activity as its predecessors finish early, and we can avoid the impact of Parkinson's Law.

THE REST OF CHALLENGE 1 -- DEALING WITH MURPHY'S LAW

But we're not yet done with the first challenge, especially the part about protecting against Murphy's Law.

We've now got a tight schedule supported by these resource alerts to assure that the critical resources are available when needed and that they can pick up the work when tasks are finished earlier than expected. The problem is that these "50% estimates" don't do too much to help us promise a final due date for the project. Through management support to allow focus, short target durations to maintain that focus, and no due dates or deadlines distracting us from what needs to be done, we've pretty dealt with Parkinson, but we've left ourselves wide open to suffer Murphy's slings and arrows. We need to protect the due date from variation in the tasks, again, especially critical tasks.



Let's look back at our original view of the task estimates -- what might be considered the "90% confidence" estimates that we have usually built our schedules on. The difference between our 50% and 90% estimates is safety. Instead of spreading it around, among the tasks, where it usually gets wasted, let's take a "whole system" view and concentrate it where it will help us. The safety associated with the critical tasks can be shifted to the end of the chain, protecting the project promise (the real due date) from variation in the critical chain tasks. This concentrated aggregation of safety is called a "project buffer."

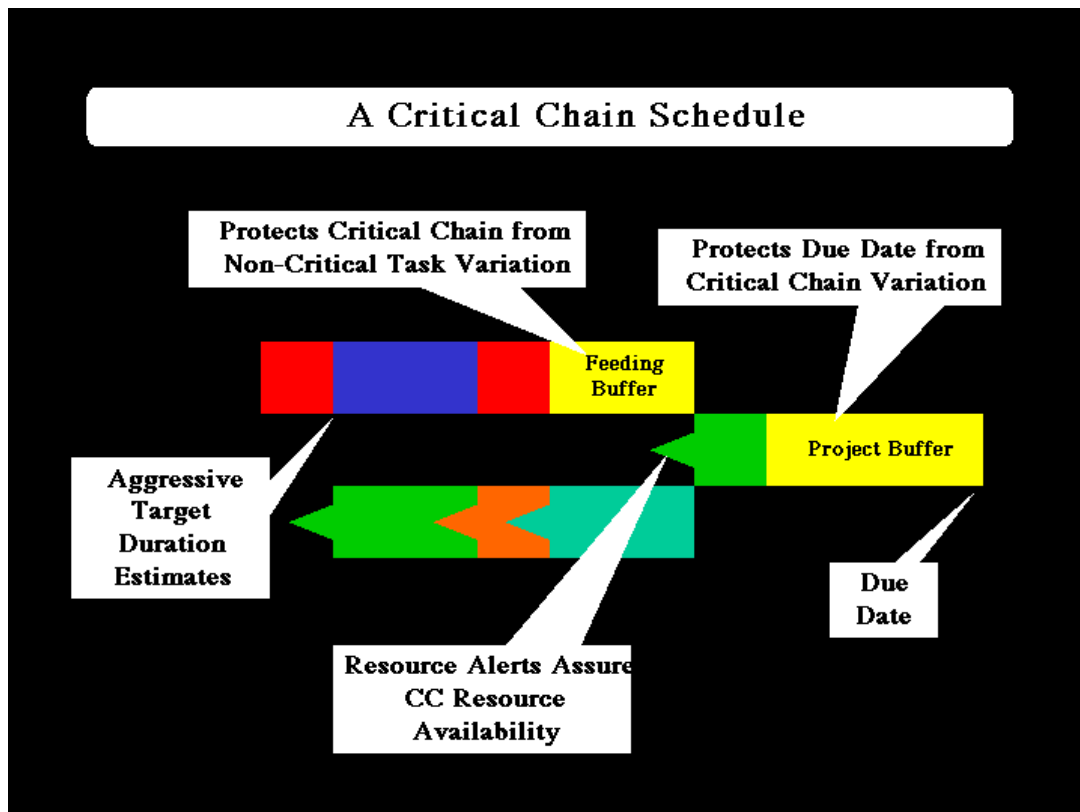
There is an additional advantage to this aggregation of safety in the form of buffers. Because the tasks' target durations are 50% confidence estimates, we might expect that half the time they'll come in early and half the time they'll be late. Since the early tasks (which we were very rarely able to take advantage of in traditional project management) will help to offset some of the late ones, we don't need all the protection that used to be spread around. So the project buffer can be smaller than the sum of the parts. I won't go into the statistics here, but we can usually cut the total protection at least in half and still be safe, resulting in a project lead-time that can be significantly shorter than in the old paradigm for a project promise of similar risk.

Now let's turn to the non-critical tasks. Let's assume that they're also allowed to focus on the task at hand and pass it along as soon as it is done--which should be a universal way of life if we really want to get projects done in a timely fashion. But we don't want to micro-manage everybody to the degree we do the critical tasks with the resource availability alerts. Yet we do want to assure that, if things go wrong in the non-critical, we don't want them to impinge the ability of the critical tasks to stay on track.

The traditional approach is to start these tasks as early as possible, and hope that the slack or float is enough to absorb the variability. It might, but then again, it might not. Why not use the buffer approach like we did with the critical chain and the project due date? In this case, concentrate the safety associated with chains of non-critical tasks (again, reduced due to aggregation) as a buffer protecting the start of the critical chain task they feed into--"feeding buffers."

Note that the feeding buffers are also relied upon to deal with resource timeliness for non-critical tasks/resources; we don't use the "work-coming alerts" because even if the feeding buffer is consumed, the worst case is that the critical tasks are delayed and maybe eat some project buffer. The feeding, non-critical tasks are two buffers away from impacting the project promise. Also, you gain more by keeping non-critical resources focused on the work at hand and to assure they finish work that can be passed on to other resources rather than interrupt them for other non-critical stuff.

We have now built a Critical Chain Schedule. (A major distinction from a schedule based on critical path methodology is the proactive approach of using feeding buffers to keep the critical chain critical up front rather than relying on reacting to a changing critical path. Another distinction, not detailed in this article, is the use of a resource-constrained critical path as the project's critical chain.)



The Critical Chain Schedule avoids expansion from Parkinson's Law by eliminating due dates and allowing us to take advantage of early task finishes. This schedule is also protected against untimely availability of critical resources by the alerts of work coming from preceding tasks. The project promise is protected from variation (Murphy) in the critical chain by the project buffer and the critical chain is protected from variation in non-critical work by the feeding buffers.

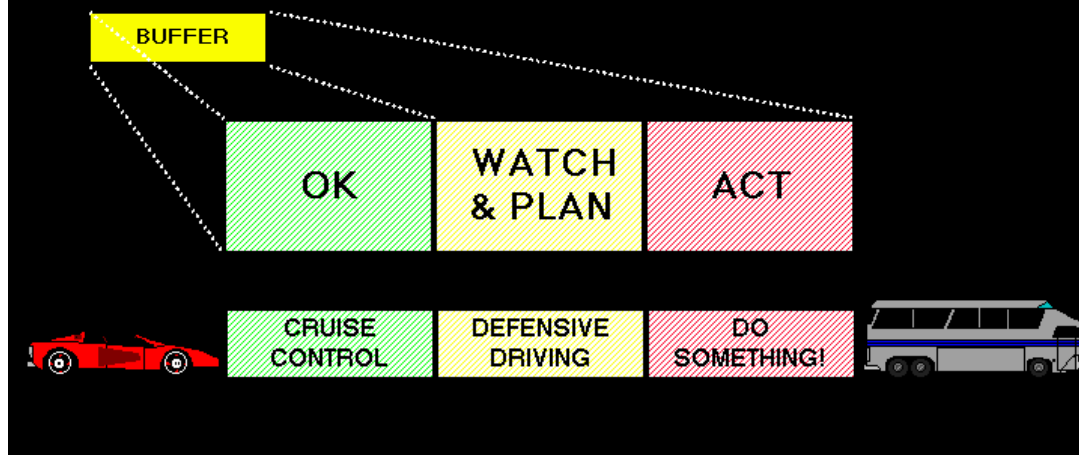
CHALLENGE 3 -- MANAGING THE EXECUTION OF THE PROJECT WITHOUT TASK DUE DATES

How can we manage the execution of a project -- how do we know what shape our project is in once it gets started, if we don't have due dates to track?

The key is the set of feeding and project buffers and a process known as "Buffer Management."

As tasks are completed, we know how much they have eaten into or replenished the buffers. Because we are now getting updated estimates of time-to-completion from currently active tasks, we can stay on top of how much of the buffers are consumed in an ongoing fashion. As long as there is some predetermined proportion of the buffer remaining, all is well. If task variation consumes a buffer by a certain amount, we raise a flag to determine what we might need to do to if the situation continues to deteriorate. If it deteriorates past another point in the buffer, we put those plans into effect.

Buffers allow focus, simplify priorities, and provide early warning regarding the health of the project. Planning and action depends on how much buffer has been consumed or replenished by task performance, which tells us when to act (and, as importantly, when not to act) to protect the critical chain and the project promise.



This process allows us to stay out of the way of the project resources if things are on track, build a contingency plan in something other than a crisis atmosphere, and implement that plan (disrupting everyone's life) only if necessary.

SUMMARY -- BENEFITS AND ACHIEVING THEM

The preceding description of Critical Chain Scheduling and Buffer Management includes, embedded in it a number of benefits that can be obtained by projects that make use of the approach. These include the following:

- An aggressive target duration schedule, along with elimination of task due-dates, minimizes impact of "Parkinson's Law."
- Buffers allow resources to focus on work without task due-date distraction and efficiently protect against "Murphy's Law" with shorter project lead-times through concentrated safety protecting what is crucial to project success.
- Resource alerts and effective prioritization of resource attention allow projects to take advantage of good luck and early task finishes while buffers protect against bad luck and later than scheduled finishes.
- Buffer Management provides focus for schedule management, avoids unnecessary distraction, and allows recovery planning to take place when needed, but well before the project is in trouble.

There are additional benefits of this approach when the concepts that underlie it are expanded to multi-project environments. While beyond the scope of this article, suffice it to say for now that the use of buffers to prioritize resource

attention will allow such organizations to allow the focus on the task at hand to speed projects in the context of multi-project programs. The Critical Chain approach to single projects allows the multi-project environment to avoid the lead-time multiplying effect of multi-tasking.

To achieve these benefits, it must be recognized that the implementation of Critical Chain Scheduling and Buffer Management is not a simple technical change of how we build and monitor projects, but requires broad management changes. Some of the significant shifts include:

- Stop spreading safety, hidden and wasted in the tasks. Concentrate safety in strategic places that protect what is important to the project from Murphy's Law. This can only happen effectively when resources trust management and project owners to accept that their tasks-- target durations are not commitments and that the buffers are sufficient to protect the project.
- Stop the behaviors that waste time in the project. Avoid task due-date focus and Parkinson's Law. Old habits are hard to break. Project managers must stop publishing date-laden project schedules.
- Avoid resource multi-tasking and the lead-time multiplication it results in. Focus on the task at hand. Management must take responsibility for protecting resources from competing priorities that drive multi-tasking.
- Account properly for resource contention. Project managers, when building project schedules must realize resource dependency is as real as task dependency when determining what is critical for the project.
- Track the consumption and replenishment of buffers. The project team must plan and act to recover when necessary, as dictated by buffer status, but only when necessary, in order to avoid unnecessary distraction of project resources who should be allowed to focus on their work.

Putting Critical Chain Scheduling and Buffer Management in place is not quite as easy as flipping a switch or turning on a new piece of software. It requires real change in how projects, resources, and priorities are managed. But if project speed and reliability are important to an organization, it may well be worth the effort to assess the potential benefits.

REFERENCES

Goldratt, Eliyahu M. : *Critical Chain*, North River Press, Great Barrington, MA. 1997

A.Y.Goldratt Institute : *TOC Project Management Workshop*, A.Y.Goldratt Institute, New Haven, CT. 1998

Newbold, Robert C. : *Project Management in the Fast Lane: Applying the Theory of Constraints*, St. Lucie Press, Boca Raton, FL.