8

$20

THINGS TO KNOW BEFORE YOU START YOUR NEXT PROJECT

By
Barry J. Adamski
Consultant / Project Engineer

"There is quite a collection of documented evidence that requirements defects are the most expensive to fix once they propagate into designs and products. In my experience with project retrospectives, I've found that around 80% of the serious defects are attributed to poor requirements.

If dealing with requirements takes the most time and produces the worst problems, then why don't we concentrate more on doing it better? There's more gold in this quality mine than in any other" Brian Lawrence, Cutter Information Corporation, Arlington, MA

Bell Labs and IBM studies have determined that 80% of all product defects are created in the requirement definition stage.

"The biggest mistakes in any large system design are usually made in the first day" Dr. Robert Spinrad, Vice President, Xerox Corporation

1. WRITE A SCOPE

What is a scope?

A scope is a definition of what is important to your project.

What does a good scope do?

> "I was teaching a class of US Marines who have the responsibility of defining the requirements for new equipment. We did several operational concept exercises, and they really got into these, but I still was not absolutely sure that they understood the importance of everyone involved in a project having a shared vision. Then, I remembered they were MARINES!
>
> "You have a vision, a vision of what it means to be a Marine," I proclaimed. "It affects the walk and talk and every action that you take. It stays with you even after you have left the Corps. I want you to define the vision for your project and share it with each participant so that the entire team is in step and on the same path."
>
> They got it!
> Ivy Hooks, Requirements Consultant, Compliance Automation, Inc. Boerne, TX

Why Scope at all?

An early scope definition keeps the project from getting off path.

A good scope has the following in it:

- Needs – These drive everything else.
- Goals – Flow from the needs (sometimes goals are the same as objectives).
- Objectives **-** Flow from the needs.
- Business case – More than just ROI and IRR.
- Operational concepts – Plain language descriptions of days in the life of the product.
- Major Assumptions – Document and keep them visible at all times for you and your team.
- Constraints – What are your budget and schedule?
- Authority & Responsibility – Are all assignments clear as to who is responsible?

The scope document should be a single-page document that all stakeholders agree upon.

Once the scope is defined, this scope must be communicated to all stakeholders of the project. A good way to determine who to distribute this to is to ask the question: Who will request changes, be a vendor to, or a customer of this project's product?

2. DETERMINE WHO YOUR CUSTOMER IS

Seems easy right? Consider though, that there are both internal and external customers. Some questions to help identify who the customer is are:

- Who will interact with this product/machine?
- Who will lose money/make money if this product doesn't sell or does sell?
- Who is paying for the project?
- Who do you report to?
- Does this product/machine go into other products?

3. OBTAIN AND PRIORITIZE THE CUSTOMER'S NEEDS AND WANTS

There are many methods to obtain the customer's needs and wants. Usually Marketing does this work. At times, though, you (engineer/project manager) may need or want to be involved in this process.

Surveys

- If you are interviewing more than one or two customers it pays to use a standard questionnaire. Using a standard questionnaire does 3 things:
  1. Ensures completeness of the survey.
  2. Ensures that the questions are asked consistently.
  3. Ensures consistent recording of answers.
- **An important question to ask yourself prior to implementing a questionnaire:** Given these answers, can you make the design decisions?
- Make sure your interviewees are a representative cross-section of your customers.
- Get the technical people involved. Market information is important and should not be left just to marketing. It can aid the technical person in making design decisions later in the project.

Camping/Fly on the wall
Pack your lunch and have your phone forwarded to your cell phone. Possibly the best way to uncover unspoken wants/needs is by camping out with the equipment or the customer at the point of usage. Spending several days simply observing how the customer(s) will use the machine can yield great benefits.

Prioritizing/Translating into "Engineer Speak"
Quality Function Deployment or building a House of Quality is a very useful tool to translate User wants/needs into design requirements. It is also used to plan which design requirements are in conflict with each other. Several books and websites are recommended at the end of this pamphlet to assist you with this process.

4. WRITE A PRODUCT OPERATIONAL CONCEPT

What is a product operational concept?

A product operational concept is a written description of what the product will be and do as written in the customer's terms.

Why create an operational concept?

- They are intuitive and easy to generate.
- They are in a language that everyone understands.
- They reduce the debate (versus writing requirements directly).
- They facilitate completeness and consistency.
- They identify user interface issues early.
- They can be used as a quick "reality check" with key customers.
- They can be used to create testing requirements for validation.

A good operational concept leads to good, complete and consistent requirements for the entire product life cycle.

5. WRITE THE REQUIREMENTS

A good requirement is:

- Needed
  A requirement is a statement of something that someone needs.

- Verifiable
  A good requirement states something that can be verified by inspection, analysis, test or demonstration.

- Attainable
  A good requirement is attainable i.e. within your budget, schedule and technically feasible.

- Clear
  A good requirement cannot be misunderstood. It expresses a single thought. It is concise and simple.
  - **Terminology to adopt**
    SHALL: for requirements
    WILL: for statements of fact
    SHOULD: for goals
    By adopting predetermined terminology you eliminate confusion.

Key question to ask for each requirement:
How will I verify this requirement is met?

6. RECORD ALL RATIONALE, ASSUMPTIONS AND CONSTRAINTS

Why do this?

> In an experiment 5 monkeys were put in a cage. A bunch of bananas was hung above a ladder in the cage. When any of the monkeys climbed the ladder and tried to take the bananas, all of the monkeys were drenched in ice water. After a while, when any monkey started up the ladder for the bananas, the other four would drag him away and beat him up.
>
> Then the experimenters began replacing the monkeys in the cage, one by one, with new monkeys. Of course, each new monkey would try to climb the ladder and the old-timers would drag him away from it. Eventually, not one of the original monkeys who had been doused with ice cold water remained in the cage. Yet, if any one of the new monkeys tried to climb the ladder, the others would drag him away from it. None of them knew why they did that, but if they were asked, they would likely say "Because we've always done it that way".

If the monkeys had written their assumptions and constraints they might have stopped beating themselves up.

<u>The cheapest, easiest and yet one of the most important things you can do is record the rationale for each requirement. No requirement should be put into a specification until the rationale behind it is well understood.</u>

Record rationale for these reasons:

- Rationale exposes bad assumptions.

  Most requirements errors come from incorrect facts. These incorrect facts come from incorrect assumptions. By including these assumptions it becomes a high probability that the false assumption will be found.

- Rationale removes unintended implementation.

> In 1932, Trans Continental and Western Air (later to become Trans World Airlines) began looking for replacements for the aging tri-motor fleet. Their original requirements read: "an all-metal, *tri-motor* monoplane, maximum gross of 14,200 pounds, fuel capacity for cruising range of 1000 miles at 150 miles per hour, carry a crew of 2 and at least 12 passengers".
>
> By stipulating that the new aircraft had to have three engines like the Ford Trimotor, they were stating implementation, a solution to a problem. Asking why they wanted three engines revealed their real requirement-an aircraft that would fly safely after one engine failed. Once Douglas Aircraft engineers understood this, they produced a two-engine plane that could fly safely on only one engine: the DC-1. This aircraft and its DC-2 and DC-3 derivatives were the first that could pay for themselves carrying passengers only.
>
> Note that, if Douglas had put three engines on the plane per the customer's original request, but two engines did not provide safe flight after one engine failed, the customer's real requirement would not have been met, even though the stated requirement would have been.
>
> Kristin Farry, Phd, Intelligenta, Inc. Friendswood, TX

- Rationale improves communication between stakeholders.

  Written rationale never goes on vacation, retires, or quits. Having a written rationale gives everyone a place to go to first to ask questions instead of to the Project Manager.

- Rationale shortens reviews.

  Better questions are asked at reviews instead of filling in assumption details every review – a significant time saver.

- Rationale maintains corporate knowledge.

  Written rationale never goes on vacation, retires, or quits. If people come into the project new they can quickly get acclimated with written rationale.

- Rationale reduces risk in defining an improved derivative product.

  When you add a new requirement, or change or delete an existing one, you take risks. If you do not understand the rationale associated with the original requirements, you may make the product unworkable or defective.

Include these items in your rationale

1. Record the *why* of each requirement.

2. Collect and record all the assumptions made.

3. Customer expected usage may drive requirements – record that assumption/fact.

4. High-level design decision – record why they are made.

7. LOOK FOR THIS IN YOUR ENGINEERING DESIGN-BUILD VENDOR

Experience in the engineer working on your project

There are many engineering companies who will have an impressive list of clients and projects. The real question is: Will the experienced engineer actually be working on your project?

A combination of electrical and mechanical expertise on staff

Some companies have only mechanical engineers. Some have only electrical engineers. The fact is most mechanisms have both an electrical and a mechanical aspect to them. So make sure, if your project/product has electromechanical properties, that the company you outsource to has that capability.

Look for signs of flexibility

You need your projects done your way. It pays to not discount the smaller design-build firm outright. Frequently the smaller company can meet your needs better than a larger concern.

8. USE THEORY OF CONSTRAINTS TO MANAGE YOUR PROJECT

Included in this booklet is information about how and why to manage projects from a ToC perspective. The author, Frank Patrick, has given permission to reprint his information here.

Who is Frank Patrick?

Frank Patrick is founder and principal consultant of Focused Performance, a management consulting, facilitation, and training resource. Through Focused Performance, Frank now concentrates on assisting organizations and individuals in building breakthrough solutions for significant results through his recognized expertise in applications of the Theory of Constraints, especially Critical Chain-based Project Management.

He can be contacted at:

http://www.focusedperformance.com/
1-908-874-8664

Critical Chain & Project Management the TOC Way

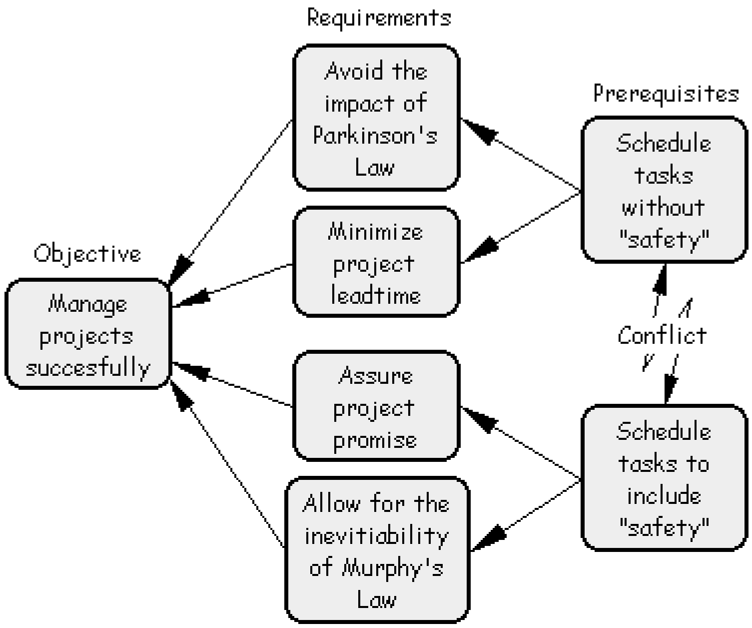
If your business suffers from . . .

- Projects over budget, not providing planned specs or scope
- Too long lead times of projects
- Chaotic hoop-jumping and midnight oil-burning to meet project due dates
- Reluctance to take on new projects

. . . then chances are good that your constraint is the way that your projects are managed.
Typically, projects are managed by focusing on the delivery of the tasks that make up the project, in the seemingly reasonable belief that if these tasks are done on time, the project will be done on time as well. But all too often, project management becomes a chaotic exercise, resulting in inordinate pressure to meet task due dates and frequent replanning of the project. Apparently, in too many cases, and for a variety of reasons, the long-established strategy of focusing on task completion does not seem to work too well.

Stuck Between Parkinson's Rock and Murphy's Hard Place

At the root of problematic systems like projects, one typically finds conflicts between two competing aspects of the system that each support needed requirements. One such dilemma is illustrated here:



Too often, this "safety" conflict is addressed by risk management efforts that provides little more than compromise or "optimization." Trim a little safety in one task, risk a little lateness against the promise. Add a little safety in another task, extend the project a little. There is never really a satisfactory compromise.
In any event, most project schedules then take these negotiated compromise numbers for task estimates and nail them to the calendar, resulting in target dates that resources shoot for to complete their tasks. As soon as a target date is set, the stage is set for Parkinson's Law ("Work expands to fill the time allotted for it.") to set in, resulting in the rarity of beating or even meeting task due dates and jeopardy for the entire project. And this is only one such conflict. There are a variety of others. Consider the plethora of conflicts and dilemmas in a multi-project environment. No wonder project management is such a challenging endeavor.

Critical Chain Scheduling and Buffer Management

"Critical Chain Scheduling and Buffer Management" is a proven "whole system" approach to project management that doesn't rely on managing a project based on a series of supposedly "safe" task estimates.

A project schedule can now be designed to protect the project due date by taking "safety" that was spread among the tasks and concentrating it where it does the most good - as buffers at the end of the critical path and where others paths feed that critical path.
Thanks to the statistics of aggregation, these buffers can be much shorter than the sum of the spread out safeties they replace, hence shortening the overall lead-time of the project.
Task behaviors are also impacted favorably; with safety reduced, the now shortened expected durations drive a sense of urgency and help to drive out resource distractions and the urge to multi-task.
Most importantly, the resulting schedule is now largely immunized from variation and uncertainty not by a futile effort to predict it at an unrealistic level of detail, but by protecting the project's promise against its inevitability.
Implementations of "Critical Chain Scheduling and Buffer Management" typically result in project schedules that can be 15-25% shorter than traditional schedules, but with considerably more reliability of the promised final project due date with less chaos and rescheduling.

Typical Results

The following are actual results available from organizations that have implemented Critical Chain Scheduling and Buffer Management - the TOC approach to Project Management:

Better Online Systems
- Connectivity solutions for IBM midrange systems

Situation

- **A project that was shooting for a due date 9 months out that no one believed**

Replanned with a Critical Chain Schedule

- **New target due date 3 months earlier than original**

Managed with Buffer Management

- **Beat the new target by a month (4 months sooner than the original questionable target)**

Saturn Development Corporation
- Construction of new car dealerships

History

- **20% overspent**
- **Project length 6-9 months**
- **Debate over whether it was on time**

After implementing TOC/Critical Chain

- **Within budget**
- **Project length 4 months**
- **On time per original plan/promise**

Harris Semiconductor
- New technology product and manufacturing facility
Project scope

- **New raw material, new automated technology**
- **New facility, doubling capacity**
- **Construction, installation, ramp-up,**
- **Focus on actual delivery of production via the TOC/Critical Chain**

Industry norm

- **Ground-breaking to first silicon - 28-36 months**
- **Time to ramp production - 18 months**

Harris results with TOC/Critical Chain

- **Groundbreaking to first silicon - 13 months**
  (3 days ahead of schedule despite losing 40 days to winter weather)
- **Time to ramp production - 21 days**

Could your organization benefit from similar results?

Critical Chain Basics

Here's a "quick and dirty" high level view of some of the major concepts involved in Critical Chain approach to project management and why they do what they do....

It all really boils down to the fact that 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. TOC/CC 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. As a matter of fact, the scheduling mechanisms provided by CC scheduling allow/require the elimination of task due dates from project plans. Its benefit is that it allows those who use it to partially avoid "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.

Project tasks are subject to considerable uncertainty, from both the unknowns of the invention process (in development projects especially) and from the universal effects of "Murphy's Law." As a result, task estimates that make up project schedules can contain considerable "safety" in them to try to allow for these unknowns when planning the project. In addition, many project organizations are multi-project enterprises, with resources frequently working across projects on more than one significant task in any particular period of time. This practice of multi-tasking, unfortunately common in many project organizations in many industries, also leads to expanded project lead times because when a resource alternates between tasks/projects, the task times for an individual project are expanded by the time that is spent on the other projects' tasks. Project resources are aware that they're in this multitasking environment and so their task estimates are further expanded (even unconsciously multiplied by factors of as much as two or three) to account for this practice in project task commitments. The combination of the effect of the multi-tasking environment and the need to cover uncertainty lead to "realistic" project task estimates that contain considerable "safety" above and beyond the actual work time required for a task, and subsequent project plans and commitments that include these expanded times.

The TOC approach addresses this expansion of project plans with two mechanisms. First, we remove the safety from the tasks, and aggregate it as "buffers" of time that are sized and placed in the schedule to protect the final due date of the project from variability in critical tasks and that protect critical tasks from variability in non-critical tasks that feed them. These buffers now allow us to shorten task time estimates to aggressive target durations, shortening the time within which resources strive to achieve their tasks. These short target durations (approximately 50% confidence estimates, whose expected overruns are isolated from the actual project commitments by the buffers), also support the second mechanism. They are so short that the resources are uncomfortable succumbing to multi-tasking or other distractions. This behavior supports the additional requirement posed by the CC methodology for management to enable resources to focus on tasks and to eliminate the multiplying effect of multitasking on project lead times. This is intuitive to many of us -- we often isolate project teams from multitasking in special task forces or "skunk works" when projects are of special importance. What the TOC approach allows us to do is apply this common sense solution to the overall project environment.

OK, so we've reduced task estimates, but we still have these buffers that include the protection that was previously spread around and hidden in the tasks. Note that I mentioned using 50% confidence estimates for the task durations. That means that, if allowed to focus on the tasks, half of the time tasks will be done in less than the target plan and half the time they will take longer. Due to the statistical nature of this uncertainty of tasks, this leads those using the TOC approach to be able to use buffers that are significantly shorter than the sum of the safety that was spread around in the previous scheduling paradigm. After all, those that come in ahead of time will replenish the buffer that was consumed by those that took longer than expected, assuring the protection of the only date that counts -- the final project due date. So with the combination of reduced task estimates due to the aggregation of safety and the reduction of buffer size, overall project plans can be typically 20-30% shorter than traditional plans with similar initial risk.

There is also another benefit of the use of the buffers, beyond protection of due date performance. They aren't just passive chunks of time in the schedule, but rather also provide the project manager and/or team with a clear indication of the health of the project at any point in time. The tracking of the consumption of these buffers provides warnings and indications of potential problems far before the project promise is in real trouble, allowing development of recovery plans in an atmosphere other than one of crisis. Once a project plan is implemented and underway, TOC's "Buffer Management" provides built-in risk management and therefore enhanced reliability of meeting the project due date, even with the shortened overall project lead time.

As a summary for individual projects, the TOC approach, by viewing the project as a whole system instead of simply as a chain of independent tasks, allows for both shorter project lead times and enhanced reliability. But as I said before, many project organizations are multi-project environments. How can TOC provide guidance for enhancing the ability of a multi-project organization to be more productive in the quantity of projects or new products undertaken and delivered?

Project and task times, due to focus and buffering, are shortened. Therefore, first we expect that the capacity hidden in and consumed by practices such as multi-tasking and task-based safety can be unleashed to simply do more work in the same timeframe. But even beyond that, the core of the TOC view of multi-project environments lies in recognizing that within a project organization, there is some resource that can be considered a bottleneck or constraint limiting the ability of the organization to do more projects. When we manage the individual projects using the TOC approach, the lack of multi-tasking and embedded safety makes it easier to ascertain the true capacity of project resources, and hence identify the constraint resource. Once the organization as a whole is managed with the constraint in mind, management attention becomes far more focused and decisions to further enhance project capacity are easier to justify and implement.

Shorter project lead times, improved reliability of project due dates, and increased capacity of the organization to take on more projects are not only predictable but have been observed in a number of organizations that have used this approach to projects in a variety of industries.

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

---

"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.

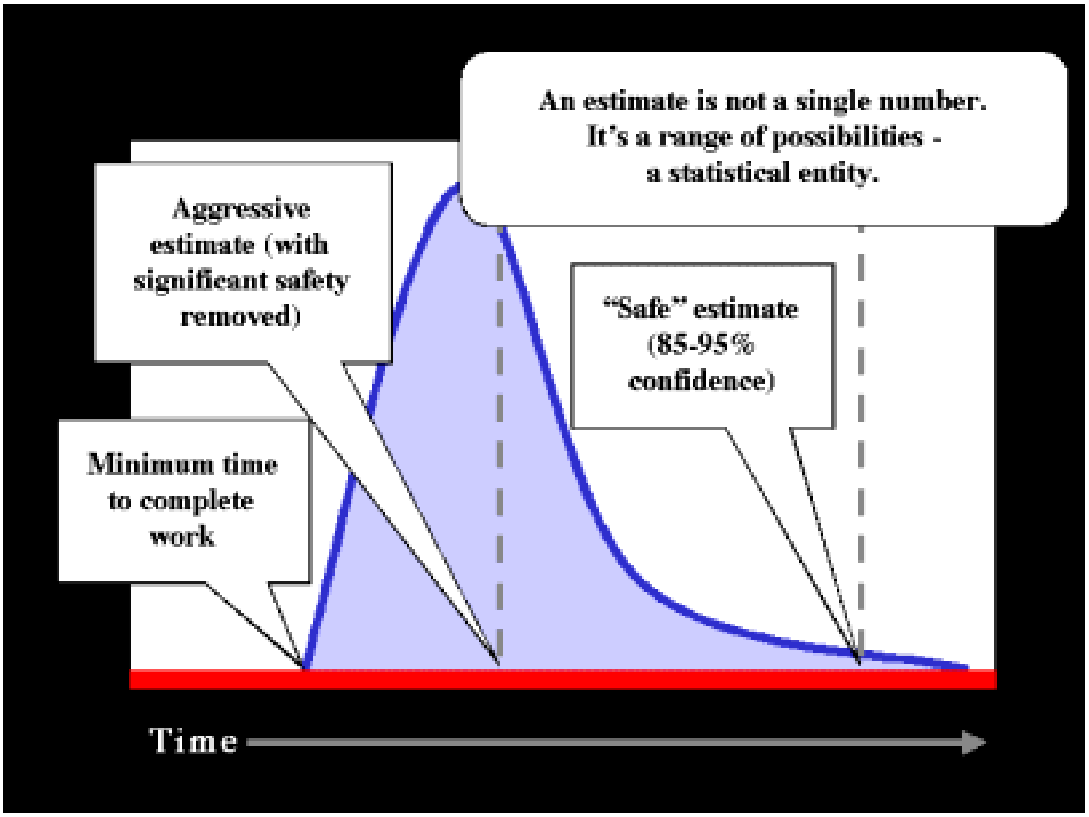
Project managers and teams need to shift their attention from assuring the achievement of task estimates and intermediate milestones to assuring the only date that matters -- the final promised due date. Safety that is typically built into tasks to cover Murphy's Law is inefficient, leading to longer 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.

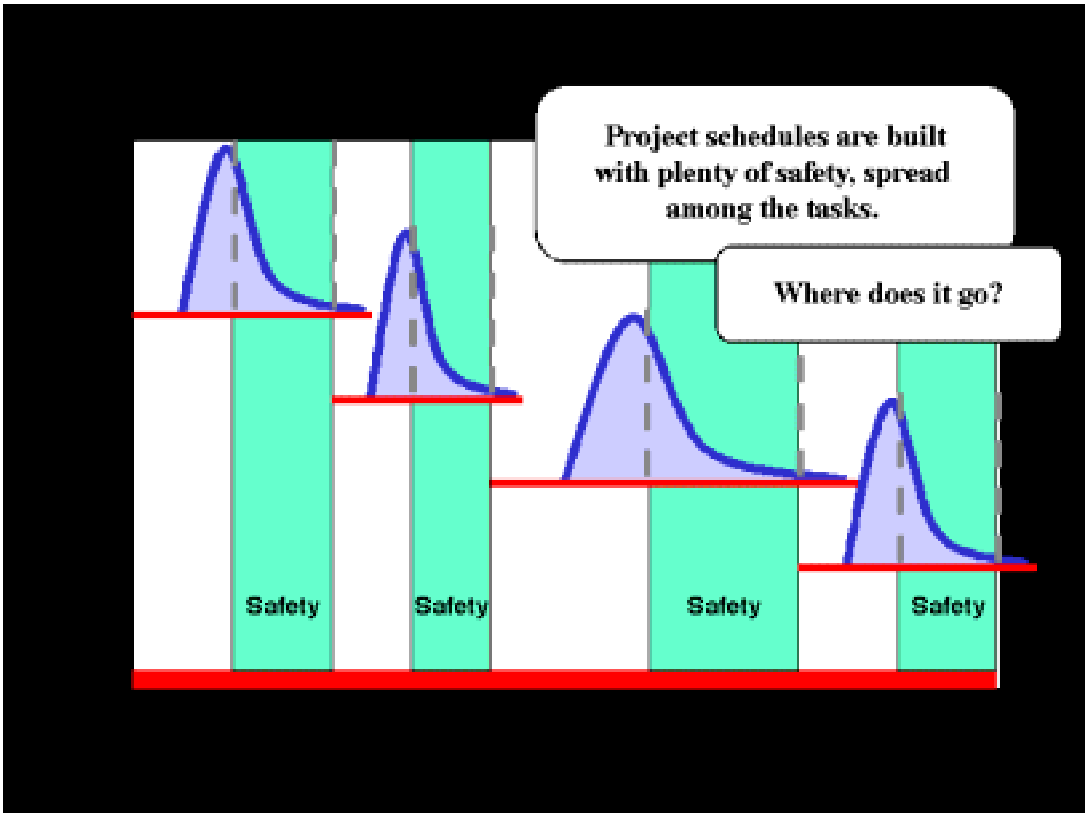
Developed through the application of the Theory of Constraints to the subject of projects, "Critical Chain Scheduling" suggests the shifting of focus from assuring the achievement of task estimates and intermediate milestones to assuring the only date that matters--the final promised due date of a project. 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.

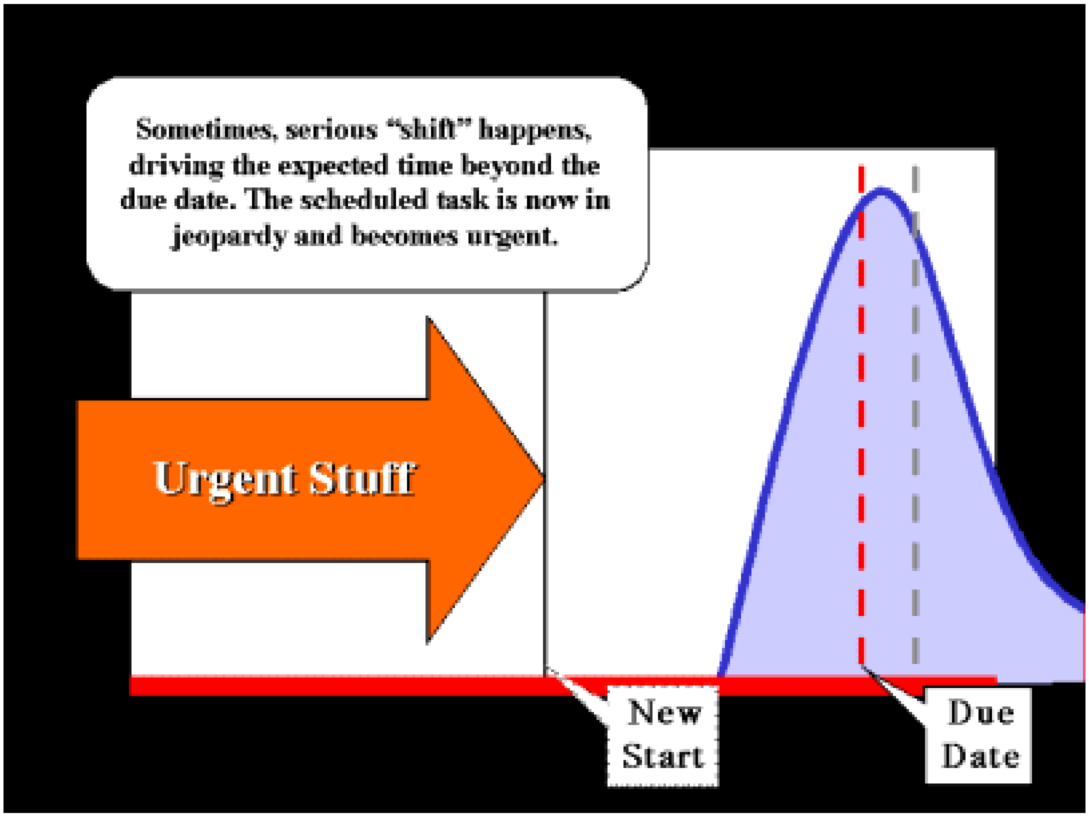


What happens to this safety? Why is it so hard to meet task deadlines and project promises? In some occasional cases, it may simply be an issue of excessive problems or erroneous assumptions overwhelming the safety, but the difficulty of bringing in projects on time is so common that there must be something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

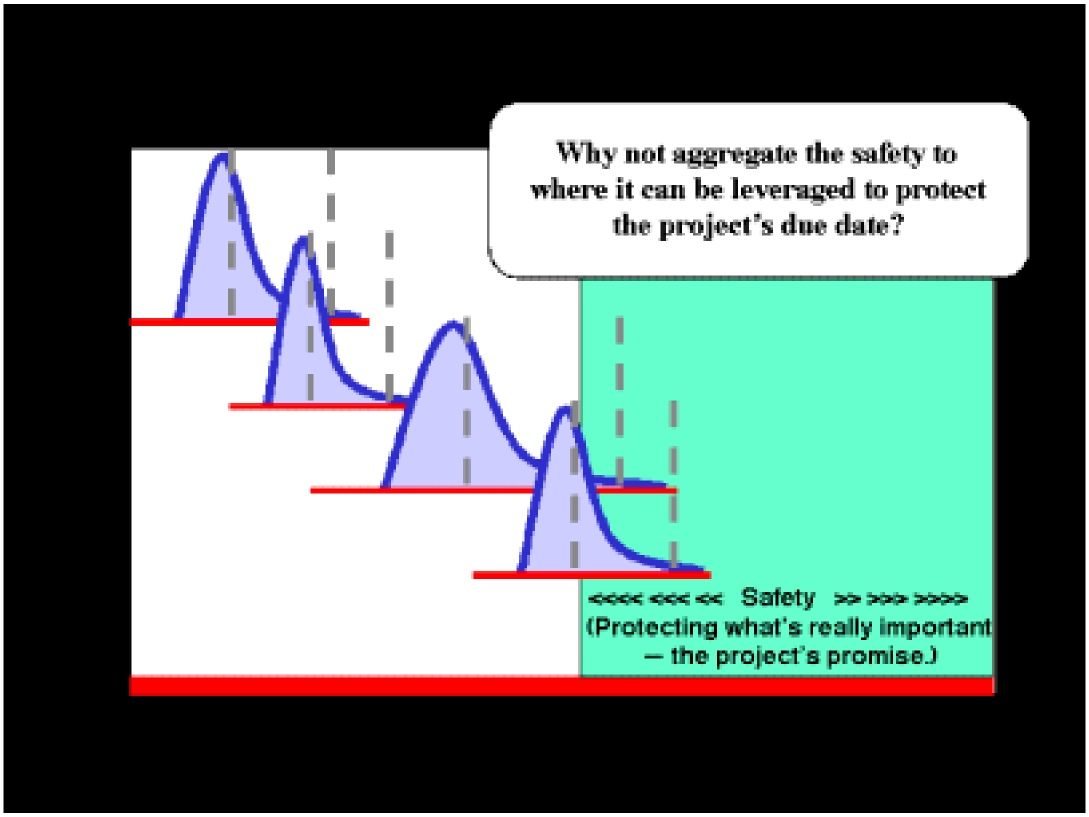
There are two simple steps required to accomplish this. Step one: Ask the resources how much of an advance warning they need to finish up their other work and shift to interruptible work so that when the preceding project task is complete, they can drop what they're doing and pick up their critical task. 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."

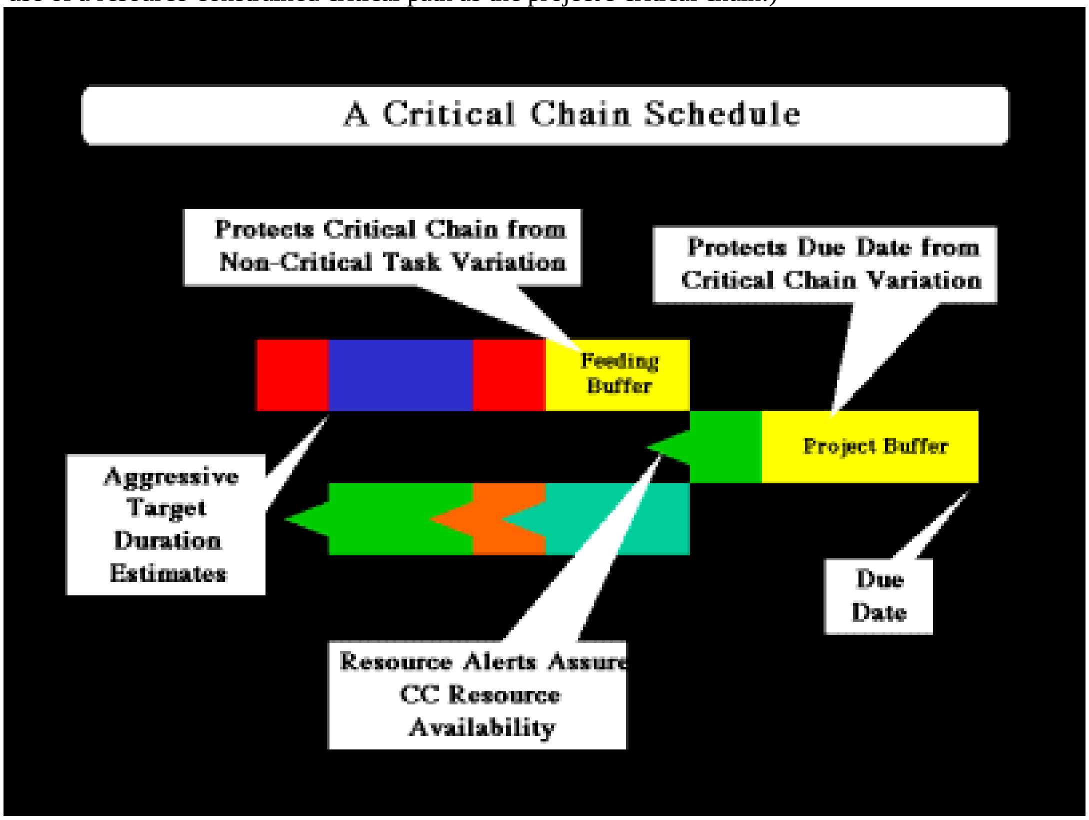
There is an additional advantage to this aggregation of safety in the form of buffers. Because the tasks' target durations are 50% confidence estimates, we might expect that half the time they'll come in early and half the time they'll be late. Since the early tasks (which we were very rarely able to 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.)



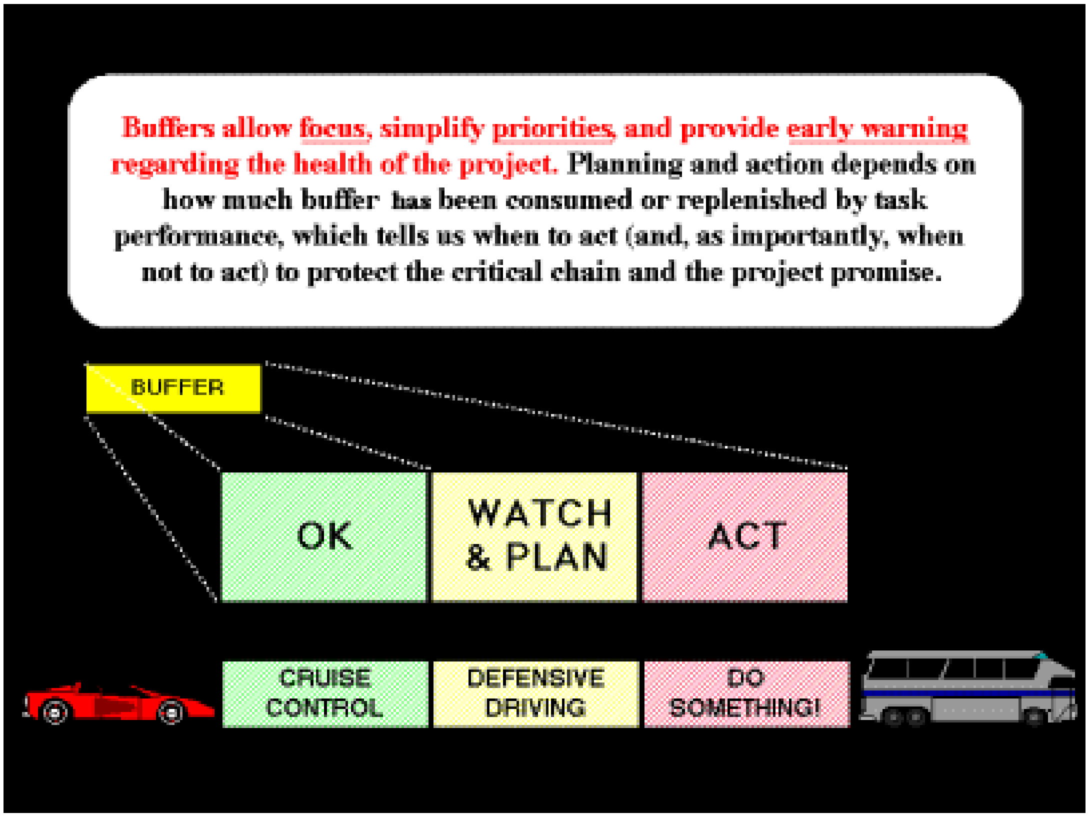
The Critical Chain Schedule avoids expansion from Parkinson's Law by eliminating due dates and allowing us to take advantage of early task finishes. This schedule is also protected against untimely availability of critical resources by the alerts of work coming from preceding tasks. The project promise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



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. 1998

Program Management -- Turning Many Projects into Few Priorities with TOC

This article was originally presented at the national Project Management Institute Symposium (Philadelphia, October, 1999)

by Francis S. "Frank" Patrick of Focused Performance

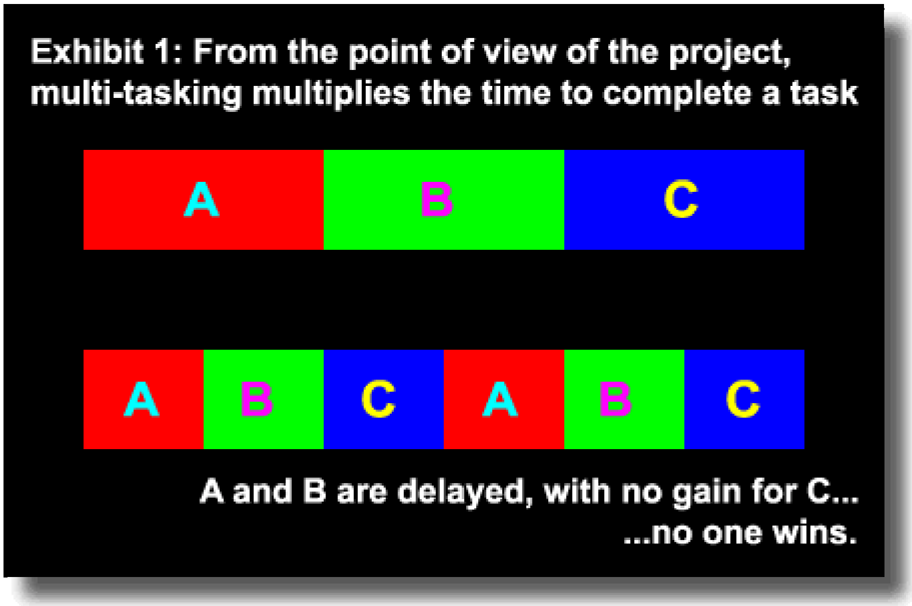
The way to get something done is -- quoting the well-known Nike footwear ad -- to "just do it." Focus on the task and get it done. For those who work in organizations that rely on programs of projects -- multi-project environments where resources are shared across a number of projects -- there are usually a lot of things that need to get done. An environment of many projects typically generates many priorities for project resources and managers alike and can make that focus difficult to achieve.

Division of attention multiplies task and project lead-time...

In an effort to take advantage of valuable new opportunities, multi-project organizations, more often than not, tend to launch projects as soon as they are understood, concurrently with existing projects, simultaneously with other new efforts, and unfortunately too often, without sufficient regard to the capacity of the organization. A common result is that the responsibility for sorting out an array of conflicting priorities often falls to project resources and their managers. One concern coming from this situation is that the resultant locally set priorities may not be in synch with each other or, more importantly, with the global priorities of the larger organization. A common result of trying to deal with this tug-of-war of multiple priorities is the practice of multitasking -- assigning resources to more than one significant task during a particular window of time -- to try to move all the projects along.

In addition, many project teams rely on early starts of projects and their paths of tasks to try to assure and achieve timely project completion. These early starts -- also driven partially by the desire to see "progress" on all open projects -- often translate to additional pressure on resources to multitask between tasks and between projects. There is pressure to get started on a new task in the in-box, but we're still working on another task. As a result, these practices of early starts and multitasking have been recognized as common practice in many organizations, and even institutionalized in project management software tools, which typically default to "ASAP" scheduling, and which offer "features" to apply "fractional resources" to tasks and to "split" tasks.

The question is, however, whether these early starts actually accomplish their desired effect. When multitasking is the result, the seemingly common sense belief that "the sooner you start, the sooner you finish" becomes questionable. True progress in a project happens only at the handoffs between resources, when the work completed by one resource allows another resource to start its work. To the extent that one project's tasks are interrupted by work being performed on other projects' tasks, the first project is delayed. The common practice of multitasking results in multiplying the time it takes to complete tasks, delaying true progress in projects. [See Exhibit 1]



If a resource divides its attention between different tasks before handing off task deliverables, all the projects involved will take longer than necessary because all of that resource's successors on each project will have to wait longer than necessary due to time spent on other projects' work. And if many resources in the organization become accustomed to working in this manner, then most projects will take significantly longer than necessary, in both their promise and their execution. The projects will also be impacted by the variability of not only their own tasks, but also of those associated with the other projects that are interleaved within them.

The pressures of these competing priorities result in the splitting of attention and energy, loss of focus, and inability to complete tasks and projects in a timely manner, or even within the time in which they were planned -- at least without heroic efforts. This is not a desirable outcome for projects that want to keep their promises, or for organizations that need to reliably deliver projects in shorter and shorter intervals.

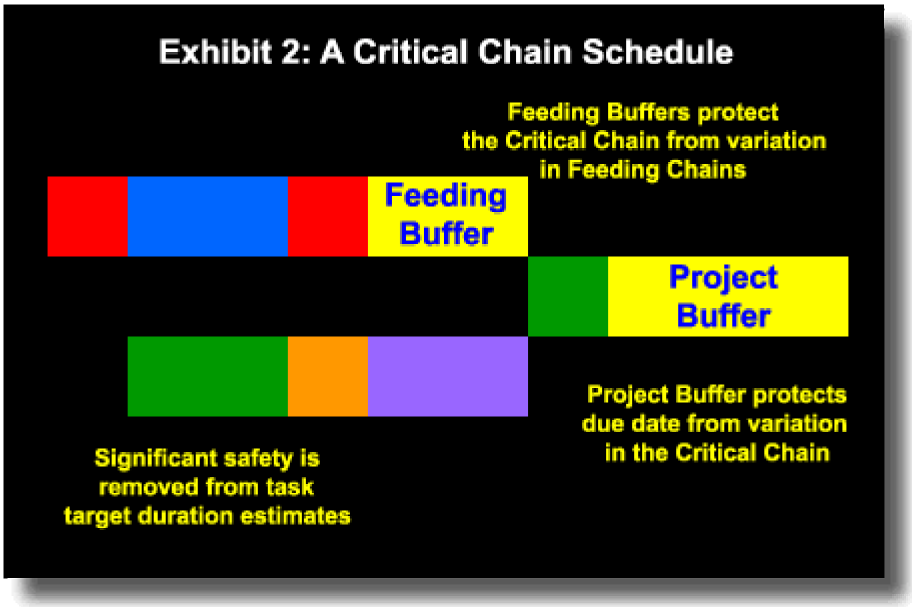
One of the key challenges of multi-project or program management therefore revolves around the avoidance of pressures on resources to multitask and the ability to assess and direct the most beneficial use of resources when there is apparent contention for their attention. To the extent that these can be addressed, a multi-project program will minimize the pain that is encountered in the interaction of projects fighting for shared resources.

Avoiding pressures to multitask...

The pressure to multitask comes from the combination of having more than one task in one's in-box and the lack of clear priority for the best use of one's attention. If there were a way of setting common sense priorities for the maximum benefit of the organization, it would make sense to all that we set aside some tasks to wait for the completion of the most critical. And if there were a way of reducing the queue of tasks waiting for a resource, there would be less need for assessing and resetting priorities. If we could systematically both provide clear priorities and minimize the queue, then the devastating impact of multitasking on projects and, more importantly, on organizational performance would be minimized.

Applying the management philosophy known as the Theory of Constraints (TOC) to the realm of project management provides a whole system view of the challenge. TOC suggests that components of the system being managed subordinate their efforts to the larger system of which they are a part. The management of tasks and the resources that perform them must subordinate to the needs of projects, and the management of projects must subordinate to the needs of the multi-project organization to which they belong. The TOC-based solution for managing single projects, whether standalone or as part of a portfolio of projects, is known as *Critical Chain Scheduling and Buffer Management.* It provides part of the answer for priority aspect of the question "What should I work on?" which, if not addressed appropriately, drives multi-tasking behaviors in multi-project environments. (Goldratt, 1997; Newbold; Patrick)

A critical chain schedule removes the pressure of artificial task due dates from the concerns of project resources. It does this by recognizing that a schedule is only a model of our expectations and by aggregating and concentrating the safety that is typically embedded in individual tasks where it does the most good, in a system of *buffers* positioned to protect the promise of the project. [See Exhibit 2]



In the real world, we expect time variation in the execution of tasks -- Murphy's Law has not yet been repealed (Patrick). Buffers are used to absorb that variation without distraction to the resource performing the task at hand, while at the same time protecting the truly critical promises of the project. The result is the elimination of meaningless intermediate task due dates and the detrimental pressures, behaviors, and practices associated with them. These include Parkinson's Law ("Work expands to fill the time allowed.") and the Student Syndrome (Delaying the start of a task due to having more than enough time to accomplish it.).

Buffers also effectively absorb deviations from the baseline critical chain model made up of target task durations from which significant safety has been removed. As long as there is sufficient buffer remaining, the project promise can, to some degree, be protected from distractions and disruptions, such as those from the need to use a planned resource on another, more jeopardized project or more critical task. If there is sufficient unconsumed buffer related to a task waiting for attention, a resource can hold off on picking it up and multitasking, and instead, maintain focus on the current task at hand until it's complete. The deliverable of the current task can be handed off before moving to the queued task, minimizing the set-down, set-up, and half-finished work that extends project lead-times when multi-tasking is the usual response.

The buffers and the status of their consumption and replenishment during the reality of project execution also provide a clear, forward-looking indication of what chain of activities is in the greatest jeopardy of delaying the promise of a project. When a project buffer is sufficiently consumed to indicate heightened risk of the project promise, then it is clear that the priority for attention by resources should be adjusted to address the tasks associated with that project. Buffers can therefore provide clear direction for the most beneficial use of a resource's focused attention.

But even with these safeguards at the individual project and task level, the pressures to jump from a task on one project to another -- to multitask across projects -- can still be overwhelming and distracting if resources are faced with an overflowing in-box of tasks clamoring for their attention.

This is due to the fact that if projects are pushed into an organization without regard to the system's capacity and capability, work-in-process (in the form of started, but unfinished projects) quickly clogs the system. The build-up of work-in-process creates queues of work that dilute and diffuse the time and attention of resources and management alike and often expand project lead times beyond the comfort zone. There is still pressure to multitask. It is therefore necessary to look beyond the individual projects, or even pairs of them, to the larger system encompassed by the organization responsible for accomplishing many projects.

Synch or sink...

In addition to providing controls and measures for individual project status or determining priorities between projects, TOC, when applied to multi-project systems, also provides guidance on assessing the capacity of such systems and related mechanisms for the synchronized launch of projects.

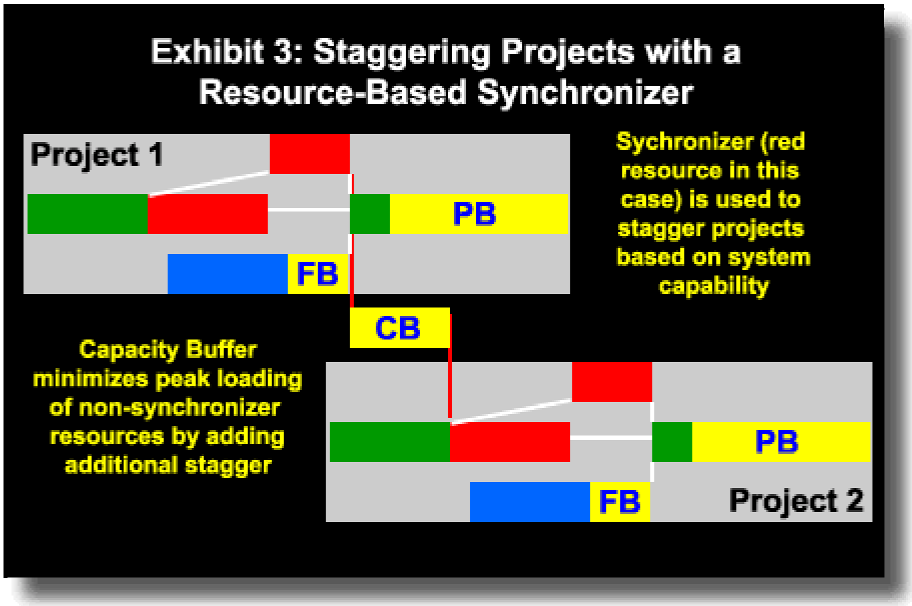
When faced with assessing system capacity, many organizations typically go into a major data-collection and number-crunching exercise in an attempt to balance the availability of all resource types with the demand on the system. To support the scheduling and monitoring of projects, however, the required process is far simpler than that usual approach. TOC tends to focus on maximizing flow of work through a system rather than balancing capacity. This higher-level view of system capacity rather than resource capacity leads to the conclusion that it is enough to keep as little as one resource effectively utilized to manage and maximize the throughput of the system. Indeed, in order to do so, it is required that other resources have sufficient protective capacity to protect that throughput. (Goldratt, 1992)

Therefore, determining a starting point for synchronizing the flow of work through the system can simply involve identifying an aspect of the multi-project system that can approximate its throughput potential. One possible candidate for this *synchronizer* might be a resource that is commonly used across projects and more heavily used relative to most other resources. (Jacob; Newbold)

The role of the synchronizer is to set the pace at which projects are launched into the system. They provide a stagger that is intended to allow overlap of project schedules, yet minimize peak loading on all resources and the pressure to multitask that is the usual result of these peak loads. Once a synchronizer has been identified, a *synchronization schedule* for the multi-project program can be put together that, combined with individual critical chain project schedules, will provide the basis for responsive, realistic and reliable project promises. To develop this schedule, projects are first assigned a strategic *precedence*. The priority against which projects will be serviced by the synchronizer is determined. When desired project commitments result in conflicts for synchronizer attention, the higher precedence project's synchronizer tasks are move earlier in time, along with the remainder of its schedule, minimizing the impact of lower precedence execution variability on higher precedence projects.

In addition to the ordering and staggering of projects provided by the synchronizer, the synchronization schedule must also take into consideration the fact that not all projects are consistent in the use of the synchronizer. This may result in occasional windows of time when the stagger is insufficient to protect other resources from peak loading and pressures to multitask. To prevent this situation, additional stagger is added between the projects in the form of a *capacity buffer*. Based on the expected variability of synchronizer work within the earlier project, the capacity buffer also provides a level of cross-project protection and time for recovery and other non-project uses for synchronizer resources.

The synchronization schedule therefore consists of the precedence ordered synchronizer tasks, capacity buffers whenever a synchronizer moves from one project to another, and natural gaps that result from the actual demand placed on the synchronizer. These gaps are important aspects of the schedule, as they allow new projects' synchronizer tasks to be interleaved among already committed projects, enabling the organization to take on new opportunities without impacting existing project promises.



The resultant rhythm of project launches [See Exhibit 3], its pace set by the capacity and capability of a commonly used and heavily loaded synchronizer resource, is well within the ability of less loaded resources to maintain. More importantly, combined with the individual critical chain schedules' systems of buffers, this synchronization schedule of projects allows resources and their projects to recover from delays and disruptions in a timely, rational, and non-heroic manner. Without synchronized project launches, the risk of sinking into a swamp of muddy priorities is too great for comfort.

Assign no task before its time...

In addition to a healthy respect and accommodation for inevitable variability in execution, and an emphasis on flow of throughput over balanced capacity, most applications of the Theory of Constraints also recognize that human behavior plays a major role in system performance. This leads to another fail-safe against non-productive multitasking built into the TOC-based approach to program management.

The particular behavior in question is the normal propensity to look ahead to and prepare for work coming down the pike. The problem with this behavior is that if a task is in a person's in-box while s/he is tending to another task, the temptation to pre-prepare for the next task could lead to distraction from the task at hand -- resulting in multitasking and delay of project progress.

In order to avoid this behavior, when developing critical chain schedules for projects, resources identified with particular tasks are done so in terms of the skill required for the task, not in terms of particular people who actually perform those skills on the task. Actual assignment of personnel to tasks by resource managers should be held back until predecessor tasks are complete and the task is ready to start. With the synchronization schedule and its resultant protective capacity now available in most resource pools, resources will wait for tasks more than vice versa. This designed situation will now provide flexibility for assignment to appropriate available people and yield maximum flow of undistracted work through the system.

Summary -- Many projects, a few clear priorities...

In PMI's A Guide to the Project Management Body of Knowledge, a program is defined as ". . . a group of projects managed in a coordinated way to obtain benefits not available from managing them individually." Most organizations that depend on the accomplishment of projects as a source of products, profits, or process improvements do so with shared resources that must be "managed in a coordinated way." In such a system, proficiency at managing single projects individually without proactively dealing with the interactions between them is not sufficient to assure the attainment of the goals of the organization. The system that really needs to be managed in most cases is greater than the sum of the single projects. It is a larger, complex system of projects, priorities, policies, and practices that guide the behaviors of managers and resources and requires consistent and coherent coordination for maximum effectiveness.

By applying the TOC prescription for multi-project/program management, an organization honors its priorities by scheduling its program through the strategically defined precedence of the synchronization schedule.

Project managers avoid unnecessary changes in priority by relying on buffers to absorb most of the normal, expected variability in the execution of tasks and projects.

Resource managers find clear direction and priority for assignment of tasks in the status of the buffers, which indicate the best use for available resources to support the promises made by the organization.

And resources have a single priority -- the current task to which they are assigned. Without the distraction of pressures to multitask or to meet false priorities of task due dates, they can concentrate on the task at and "just do it," do just it, and do it justice to assure a quality hand-off, successful projects, and maximum throughput for the organization.

References:

Goldratt, Eliyahu M. 1992. *The Goal, 2nd Revised Edition*, Great Barrington, MA: North River Press.

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

Jacob, Dee. 1998. *Introduction to Project Management the TOC Way -- A Workshop*. New Haven, CT: The A.Y.Goldratt Institute.

Newbold, Robert. 1998. *Project Management in the Fast Lane.* Boca Raton, FL: St. Lucie Press.

Patrick, Francis S. 1999. "Getting Out From Between Parkinson's Rock and Murphy's Hard Place." *PM Network 13* (April): 57-62.

Top 10 Sources of Project Failure

Do any of these conditions exist in your organization?

Do you see the possible linkages between them and poor project performance? Think about whether you are doing anything to address these potential roots of problems. If not, then you might be stuck at current performance levels

1) Failure to appreciate the impact of a multi-project environment on single project success. (**More...**)

1 a) Trying to put 10 pounds of projects through a 5-pound pipeline in a multi-project environment.

1 b) Wasting of resources through dedication to projects, making them unavailable to support other projects.

1 c) Failure of management to provide real guidance on priority of projects before they are planned and promised.

1 c1) As well as the flip side, ignoring rational plans and promises for perceived, but questionable, priorities. As an explanation of this, IMHO, project priorities are part of the initiation phase. Once promised and launched, all project have equal priority **--** to complete when and how promised **--** and deserve attention only proportionate to the threats to that promise.

1 c2) Another flip side regarding priorities **--** failure of management to kill projects when their reason for existence goes away.

2) Irrational promises made due to a failure to take into account the variable nature of task performance. (**More...**)

3) Irrational promises made due to a failure to take into account the statistical nature of project networks. (**More...**)

4) Insufficient identification of dependencies necessary to deliver the project. (**More...**)

5) Focus on (and active management of) only a portion of what should be the full project **--** a true bottom-line value adding outcome for the sponsoring organization. (**More coming...**)

6) Reliance on due-date, train-schedule, and actual-against-budget-to-date performance to drive project performance, resulting in the wasting of any safety included in the project (to account for 2 and 3 above) and in the effects of Parkinson's Law **--** Work will expand to fill (and exceed) the time allowed. The whole concept of "time allowed" is a major culprit. (**More...**)

7) Wasting of resources through underutilization because they aren't the "best resource" for the job. (**More coming...**)

8) Wasting of the "best" resources through over-utilization, multi-tasking, and burn-out. (**More coming...**)

9) Delivering original scope when conditions/needs change. Flip-side: accepting changes to scope without sufficient analysis of impact on the project (or on other projects). (**More coming...**)

10) Multi-tasking, multi-tasking, multi-tasking, multi-tasking, and multi-tasking. Commonly thought of as a key problem in multi-project environments, where resources are expected to address tasks from different projects in a coincident time-frame, multi-tasking also impacts single project durations (and wastes safety) when dedicated resources are expected to wear several hats.

The Sooner You Start, the Later You Finish

William wrote...

>-- "But there are all kinds of risks and costs of starting things as early
>as possible." Not true. There is no cost to starting early. The risk is
>with NOT starting early. Good practice has ALWAYS been to schedule all
>activities to start as early as possible -- that's why it's the default in
>most scheduling software!

Allow me to suggest some risk/costs of starting early, in increasing order of concern.

- A negative hit on cash flow from spending before one has to.
- Dilution of focus on the part of the project manager and team from what must be done to what is being worked on because we can. (Possible confusion on project progress if too much weight is given to work completed unnecessarily.) Since most projects consist of parallel paths of work, any work done sooner than necessary will just have to wait for merging paths anyhow. Kind of like undesirable work-in-process inventory in production.
- The possibility of doing work that isn't a predecessor to, but can be influenced by learnings in parallel paths of the project, leading to potential wasted effort and possible rework.
- For multi-project environments, where several projects are pursued with a shared set of resources, earlier starts than necessary will tend to drive multi-tasking without good reason, adding otherwise unnecessary effort, time, and waste to all projects involved.

The first three are real enough, but I'd like to focus on the fourth as the most compelling reason. Many project environments are multi-project environments, dependent on shared resources that a variety of projects are calling for. There is often all kinds of pressure not only to start tasks early, as William suggests, but also to get whole projects started as soon as funding is available. One common way to manage resources in this kind of environment is to use the idea of the fractional headcount -- half time on one project and half time on another, and unfortunately, often another half time on a third project. In every Critical Chain workshop I've given, this mode of operation has been readily recognized by the participants. It allows the organization to get projects started, and after all, the sooner you get started, the sooner you finish. Right?

Wrong.

If you start projects early, and utilize the common practice of multitasking to do so without pulling resources completely off of other projects, then all the individual projects suffer from extension of lead time AND the entire organization suffers from a loss of "project throughput."

The reason for this should be quite intuitive. If you interrupt work on one project prior to handing it off to work on another project, one project sits idle while work is performed on the other. This results in longer lead times for all the individual projects involved.

The second benefit, additional throughput, comes from the combination of avoiding multitasking and the staggering of projects by their use of a heavily loaded common resource.

Tough to draw it with text, but if there are three similar projects:

a1a1a1b1b1b1b1b1c1c1
a2a2a2b2b2b2b2b2c2c2
a3a3a3b3b3b3b3b3c3c3
widespread multitasking would result in:
a1 a1 a1b1 b1 b1 b1c1 c1
 a2 a2 a2b2 b2 b2 b2c2 c2
  a3 a3 a3b3 b3 b3 b3c3 c3
completing 3 projects in the time shown -----|

(Not counting time lost to "context switching.")
Focused effort and "constraint" scheduling (staggering the projects based on the use of the "b" resource in this case) would result in:

a1a1a1b1b1b1b1c1c1
a2a2a2b2b2b2b2c2c2
a3a3a3b3b3b3b3c3c3
a4a4a4b4b4b4b4c4c4
a5a5a5b5b5b5b5c5c5

Starting projects 2 and 3 LATER results in them finishing SOONER. Also, there are almost five projects COMPLETED in the time needed for three projects with the same level of resources over that timeframe. Plus the organization would see the benefit of the original three projects in an average of less than half the time required in the multitasking mode.
I know these are simplistic examples, but I believe appropriate for this limited text-based medium, and hopefully sufficient to make the case for the desire to avoid multitasking and early starting of projects.
But back to the issue of starting project tasks early...
Sometimes a set of project schedules will result in resource contention across projects. Even if resources are leveled up front to try to avoid multitasking, "stuff happens" and tasks shift in time, resulting in unplanned contention. If a schedule is designed with tasks happening as early as possible, as suggested by the software's default, these resource contentions are sometimes unnecessary for the success of the individual project, take management attention and effort to try to resolve, and worst of all, have the potential of impacting not only the current projects but the throughput and timing of other projects as well.
I'm not suggesting that non-critical activities start as late as possible, just that they don't start as early as possible. The critical chain approach suggests they start "early enough" to avoid impacting the ability of the critical tasks to proceed, and that "early enough" is defined by the feeding buffers.
Keep in mind that "early as possible" may sometimes not be "early enough." The slack that results from "early as possible" starts in critical path schedules is simply a mathematical outcome of the network of tasks and may very well not be sufficient to protect the ability of the critical tasks to make progress. (That's what happens when non-critical things become critical and why critical paths change through the life of the project.) The buffers, on the other hand, are designed/sized to provide that necessary protection.

Multi-tasking Multiplies Lead Time

Deb wrote:

>...snip ...Most of the
>members of my project teams (I have 2 projects right now) are in the
>application areas and are not directly associated with the Y2K project team
>that has been assembled -- I have their time for anywhere from .25 FTE to
>.5 FTE. I was told that none of them could be allocated any more than this
>due to their responsibility to support production. There are usually only
>1 or 2 people supporting each third-party application. In addition, due to
>the physical structure here, they are not even in the same building as I
>am.

The idea of assigning a half or a quarter of a headcount to a project is a good way to start down the slippery slope of multitasking. When I talk about multitasking, I mean bouncing back and forth between two or more tasks before they are done and handed off. By task, I mean a chunk of work, that, when complete, is handed off to someone else for additional work.

People can easily be involved with more than one project without multitasking, as long as they complete the individual tasks they are involved in before moving to the other project's task. If, however they bounce back and forth, both projects suffer due to the time spent on the other one.

Let's say we have three tasks (A, B, & C) on three projects, each taking 4 time units of effort.

AAAA
BBBB
CCCC

If multitasking is in effect, that means that the resource will bounce back and forth at the beck and call of the three project managers. Let's make it simple and assume that each task gets half completed before moving off to satisfy the needs of another project status review.

The work would look like:

AABBCCAABBCC

instead of:

AAAABBBBCCCC

if the resource is allowed to focus on one before moving on to another. Actually I'm being kind here, ignoring additional setup and rework that will further expand the first example. But even without it, while yes, all three tasks were done in the 12 time units, from the point of view of the individual projects, each task actually took 8 time units instead of 4, resulting in the expansion of the project lead time and delay of downstream tasks relative to the start of A, B or C. If all resources on a project are working in this mode, than it's an easy leap to the statement that many of our project plans could very well be at least twice as long as they need to be.

The individual projects would actually be better off if they didn't pressure the resource to work on their task and allowed them to focus on the task at hand!!!

Project Portfolio Management -
The First Cut is the Kindest Cut

One of the common problems faced by project-oriented organizations is having too many projects relative to their capacity. Therefore, one of the first things that needs to be done is to determine what can be done is to determine what should be done . . . and what should not be done.

An appropriate priority process can be described as one that identifies projects that best support the organization's efforts for increasing its ability to achieve its goals. Assuming that we are talking about a for-profit organization, the goal is probably defined (or at least measured) chiefly in terms of making (more) money now and in the future.

Every organization, at any point in time, has some primary source of limitation - a primary constraint - impeding its ability to do more of what it's trying to do. Even more important, and useful to us as I will point out later, is the fact that this constraint colors the policies, practices, and resulting problems found throughout the organization.

One false step taken by many organizations is undertaking "improvement" efforts (and the projects that implement them) spread all around the organization, addressing the local symptoms and strengthening all of the links in the "value chain." (It's only fair. We don't want to shortchange any particular area. Otherwise they won't support our budget next year.)

The problem is that this is a significant waste of time and attention, because only those efforts that address the constraint (the weak link of that value chain) are worthy of being called real improvements. Improvement is not related to local changes in the individual links, functions, or processes, but only to the ability of the organization -- the entire chain -- to lift more weight, so to speak. Improvement only comes from the ability of the organization to acheive more of its goal(s).

Understanding this aspect of organizational systems and this definition of improvement, it becomes obvious that the projects that should be highest on the portfolio list should be those that address or support the major constraint.

The first step, then, is to identify the organization's primary constraint and an appropriate direction for an organization-wide strategy for dealing with it. Then, identify the projects that are most directly related to that strategy, and to the ability of other aspects of the organization to support it. Note that by focusing on organizational constraint issues, this ranking is more of a qualitative, intuitive process more than a typical mathematical matrix. Obviously, a comparison of ROIs may come into play for fine-tuning, but if you start with addressing the constraint, the returns will be real and significant, and you won't go too far wrong.

My suspicion is that of most projects in a typical portfolio, there are actually very few that address more than just local cost-cutting efforts of questionable value; very few that actually support growing the ability to attract demand from the market and deliver against it reliably and profitably. Without an understanding of your major constraint and some rational direction for addressing it, any simplistic effort at rating or ranking your list of projects will probably fall short of what is needed.

The good news is that getting a handle on the constraint and a strategy for dealing with it is easier than it sounds, as long as the right people are involved. Since I would expect that unexamined portfolios cover most if not all of the functional areas of the organization, those areas all have an interest in the process for determining the priorities. Even more than that, the management team has the *responsibility* for being involved. A Program Office or Project Management Office can provide a process for understanding the constraint and prioritizing projects, but the top management responsible for the functions and processes have to provide the intuition and input regarding what is important to them.

TOC provides a process for quickly identifying the organization's real constraint for and aligning the individual functions with the efforts required to address it. This process is based on identifying the top functional problems and putting them in context with that constraint.

Remember that the existence and the nature of the constraint is linked to most major issues faced throughout the organization. Once management understands how the constraint is at the root of their own individual problems, they will quickly come to agreement on what projects are needed to address it, and more importantly which projects are *not* appropriate efforts for the attention of scarce resources. What better way is there to get an group of people behind an effort than to bring them to understand what is in it for them individually are solutions to their major headaches.

The process for doing the above can usually be facilitated with about a week of effort, assuming a typical participating management team of 8-12 executives with some minimal pre-preparation. Once an appropriate mind-set is developed through an introduction to the concept of the constraint as the appropriate center of management attention and to how it can be applied to a variety of typical organizational functions and processes, the actual process of understanding the organization's situation can start.

The first actual step is aimed at understanding the individual pains felt by the management team and the functions they represent, and from that understanding, dig down to identify the core, constraining issue. Frem there, directions for dealing with that constraint and for the original individual problems that align to it, and through it, to each other. It is expected that a number of projects in the incoming portfolio will surface as supportive of these solutions, and a number of others will be obviously seen as of questionable value. After that, we assess any remaining "borderline" projects, identify precedences that will also guide priority, and finalize buy-in to the list as an appropriate portfolio for meaningful organizational improvement.

As a result of this effort, management participation assures appropriate input on what needs changing and informed intuition for developing what might be considered strategic directions. It also yields clear and consistent messages regarding the appropriate project priorities across the organization.

Critical Chain and Risk Management

Protecting Project Value from Uncertainty

Francis S. “Frank” Patrick
Focused Performance, Hillsborough, NJ, USA
fpatrick@focusedperformance.com

Protecting the value of a project involves dealing with the uncertainty that will be associated with its delivery. The role of Project Management is to assist in turning uncertain events and efforts into certain outcomes and promises. If this is the case, then the primary process associated with project management should be that of risk management. How other processes, such as scope, schedule, and spending management support risk management is therefore critical for successful project management and for maximizing the value of our project-based efforts. One of the more recently introduced project management methodologies has at its core a focus on the management of uncertainty and risk.

Critical Chain-based project management has received considerable attention in the Project Management community since it was broadly introduced in Eliyahu M. Goldratt's book, **Critical Chain** (Goldratt, 1997). Most of this attention has been focused on the areas of schedule development and management. But the details of the scheduling methodology – the critical chain versus the critical path, just-in-time starts replacing as-soon-as-possible starts, the eschewing of task due dates and use of buffers of time to protect the project's promise and monitor its progress – are only means to an end. Or rather, ends – speed and reliability of project performance unencumbered by conflicting pressures and behaviors. And reliability of project promises is as much a result of a methodology's ability to support effective risk management, as it is a result of effective planning and scheduling.

Recognition of uncertainty and its associated risk are at the core of the initial stages of developing Critical Chain schedules. The emphasis on dependencies in the usual approach to developing a project network for a Critical Chain schedule helps to avoid risks of missing interactions of different parts of the project. The use of 2-point estimates to assess and address the early view of schedule risk associated with task uncertainty sets the tone up front for the appreciation of risk in the real world. In addition to task uncertainty, iteration uncertainty (a topic not written of much to date in the Critical Chain literature) can also be taken into account in the sizing of Feeding and Project Buffers. These resulting buffers themselves become a highly visible and direct assessment of the schedule risk associated with the project as a whole.

Critical Chain-based project management is more than just Critical Chain Scheduling and Buffer Management. The genesis of Critical Chain in the Theory of Constraints (TOC) has yielded a holistic view of project management that provides effective risk-focused approaches not only to scheduling and control, but also to initial scoping and planning, effective resource behaviors, and minimizing cross-project impacts. These key aspects of the methodology have a range of implications for the support of basic risk management processes and outcomes, including identification and assessment of risks, response development – bit it avoidance, mitigation, or acceptance, and guidance for response control (Pritchard, 1997).

PROJECT PLANNING – DEPENDENCIES AND DURATIONS

No matter how good a project schedule is or how well resources perform in the execution of tasks in that schedule, if critical dependencies associated with the project are not included in the description of the effort, they represent considerable risk to delivering project value.

The Dependency Network

The primary aspect of planning in a Critical Chain environment is a process known as Network Building. It is a multi-pass approach designed to assure that no key dependencies for the project are missed. Like all effective project management planning processes, it starts, as Stephen Covey might say, "with the end in mind." It requires careful consideration of what the project is about, emphasizing identification of the true value-generating aspects of the project. The TOC origins of Critical Chain provide a basis for this clarification, in terms of providing a focal point in the relationship of the project to the project owner's constraint and the contribution of the project to an enhanced ability to achieve (more of) the organization's goal. From this analysis comes necessary clarity of objectives, deliverables, and success criteria, assuring that everyone is on the "same page" regarding success of the project.

Once the end is understood, Network Building quickly shifts to a focus on task dependencies required to get there. The clear definition of deliverables serves as a high level WBS, but rather than continue developing the individual hierarchical branches of a WBS, Network Building shifts to dependency identification. To the extent that projects are highly interdependent efforts, this emphasis on what is needed to develop a handoff is a more straightforward way of building a holistic set of dependencies than trying to search across the lower levels of branches of a traditional WBS. This also assures an emphasis on deliverables and handoffs through their identification as necessary inputs, with clarity further enhanced by a strong preference for verbose task descriptions (Jacob, 1998).

Once the first pass of major dependencies, from end to beginning have been developed, it is addressed for identification of the minimum resource capability needed for task completions. Very often, this emphasis on "minimum capability" helps identify additional supporting task dependencies, as assumptions about the use of masters versus journeymen and apprentices are uncovered. Once resource identification is complete, if they haven't already been involved, the next pass in Network Building is a review of the task structure by representatives of those resources, highly useful for further catching missed dependencies and for assuring clarity of expected task outcomes.

Network Building and Risk Identification

The emphasis of Network Building in a Critical Chain environment is on clarity of task inputs necessary to support that task's deliverables. The resulting discussion of input requirements are directly related to risks associated with the ability of that task to do what needs to be done for its required output. The "backward" building of the network assures that outputs are understood before defining inputs. This focus on dependencies is, in effect a focus on risk, since missed dependencies in plans and schedules are a serious source of risk. The repeated questioning of "what do you need?" followed by "is there anything else that is needed?" serves to trigger thought of things that could go wrong, i.e., identification of potential risks in delivering task outputs.

The iterative process (initial identification of task outputs and inputs, through resource identification and review, to estimation of durations and iterations) provides a series of "safety nets" enhancing the chances of catching more missed dependencies and risks.

Network Building and Risk Avoidance/Mitigation

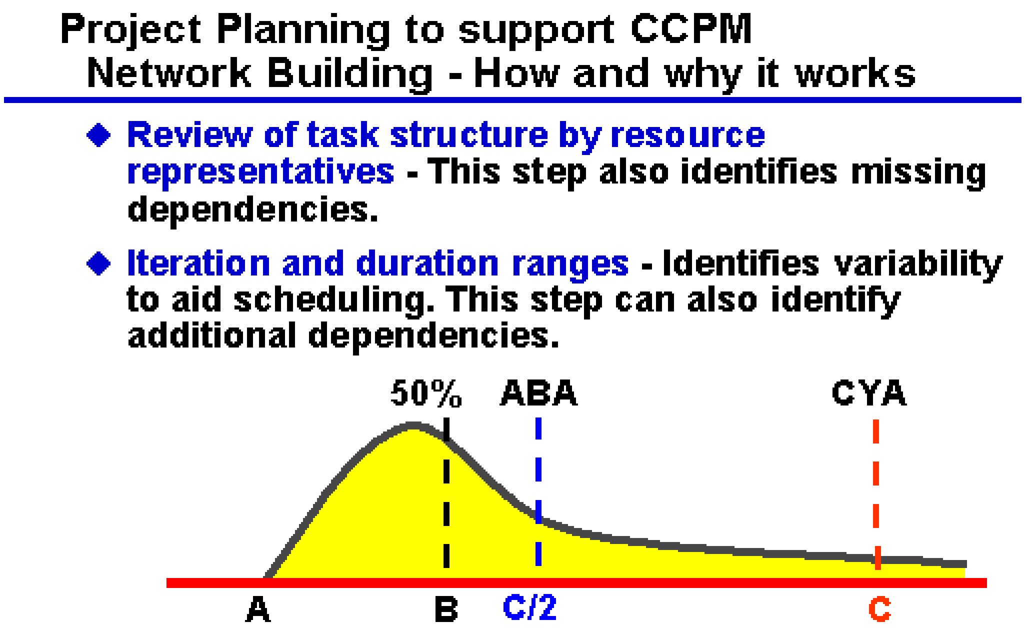
Any effective planning process is about the identification and inclusion of necessary handoffs. These handoffs are the linkages of the chain of tasks – they serve as inputs to some tasks and are developed as outputs of others. The plan – the dependency network – is simply the sum of handoffs that need to occur to overcome obstacles on the way to the project's objective and to minimize the effect of potential pitfalls along the way. To the extent that careful consideration is given to the completeness of necessary inputs, identified risks can be avoided or mitigated by adding additional tasks to the network.

In Network Building, the emphasis on input identification rather than on a flow of tasks or on isolated legs of a WBS helps to assure that risks of missed dependencies are avoided. It is far easier, once one's required outputs are identified, to come up with the full complement of necessary inputs, rather than try to guess what one's successor needs, especially when that successor is in some separate leg of a WBS.

Too often, plans include assumptions regarding the existence of necessary inputs. The incessant (some might say obsessive) focus on whether all identifiable inputs are sufficiently provided for in the network goes a long way to avoiding and mitigating risks that might have otherwise been buried in those assumptions.

Durations and Iterations

The final step in Network Building is the development of range estimates for both task durations and iterations. Critical Chain Scheduling utilizes a 2-point estimate, for both durations and iterations. Avoiding the idea of the oxymoronic "accurate estimate," the Critical Chain approach explicitly accepts and takes into consideration the reality of variation and uncertainty associated with every project endeavor. With knowledgeable representatives of the appropriate resources involved, it is critical to understand what might happen in the event that Murphy's Law strikes, and what could happen if the task in question "gets lucky." This is applicable to both durations of particular tasks, as well as to the number of iterations associated with dealing with things that are unknown up front.



To this end, resources are first queried for a safe estimate – one in which they have a high level of confidence, and are willing to consider a commitment. This defines the upper end of the possible requirements of the project components in terms of time. Once this upper limit is initially established, a second "aggressive but achievable" estimate is solicited – one that reflects a near "best case" situation that is "in the realm of possibility" if things go well in the performance of the task in question.

Two-Point Estimates and Risk Assessment/Avoidance/Mitigation

Schedule and cost risk assessment are inherent in Critical Chain's 2-point duration and iteration estimates. Once basic dependencies are identified in the Network Building process, the uncertainty and potential variation associated with individual tasks and groups of tasks are the next link related to the risk of keeping project promises and delivering desired value. Even if identified, mitigated, or avoided through additional tasks in the network, task delivery is still subject to technical or performance risk.

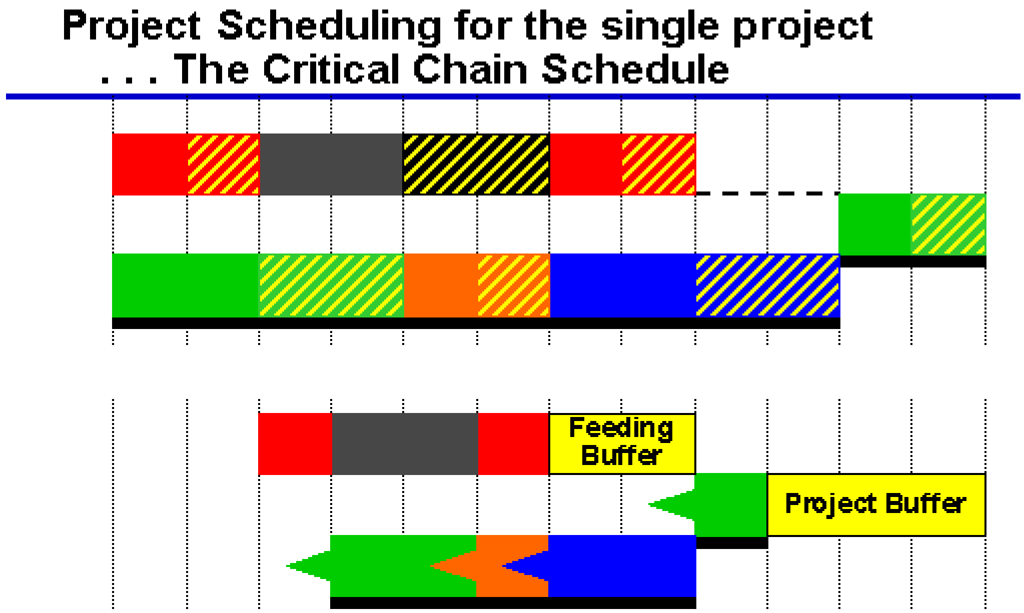
In addition to serving as the basis for turning the dependency network into a Critical Chain schedule, the 2-point estimation process is an excellent vehicle for further understanding risks and ways of addressing them. The first, commitment-level estimate should reflect the inherent task or iteration risk associated with the piece of the project in question. The difference between the larger and smaller estimates is directly related to the assessment of necessary "safety" associated with task estimates. The amount of safety necessary, relative to the aggressive but achievable estimate will highlight the level of risk associated with the task. The reasons for that safety, which should be a piece of the estimation discussion, will provide opportunities for identifying additional inputs and tasks that can serve to rationally reduce either or both of the estimates, or to help assure that they won't be exceeded in execution.

PROJECT SCHEDULING
– INTEGRATIONS, VARIATION, AND RATIONAL PROMISES

Some of the major beneficial effects of the Critical Chain approach come from the linking of scope and time management to risk management. The intimate interactions of these processes in the Critical Chain approach make them difficult to pigeonhole in the taxonomy of the traditional project management body of knowledge (Duncan, 1996), and therefore, they can easily be overlooked. The conduits for much of this connection are found in the development and use of schedule buffers and in the process of Buffer Management.

The Critical Chain Schedule

A Critical Chain schedule takes advantage of the 2-point estimate process to translate the dependency network into a reliable project promise. Reliability comes first from feasibility assured by explicitly including resource dependencies as well as handoff dependencies in the determination of the critical chain/path of the project.

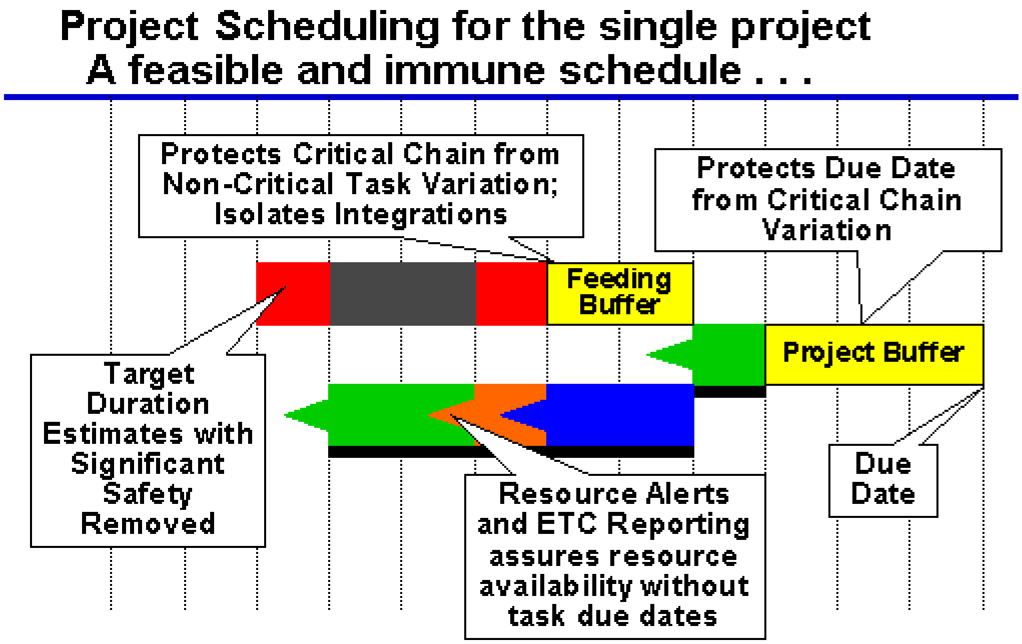


Secondarily, the two estimates developed in the planning process are used to aggregate and concentrate safety where it will do the most good to protect the project's promises and its intended value. The body of the schedule – the network of tasks and resources used to identify the critical chain – makes use of the smaller of the two estimates. The difference between the "safe" estimate and that "aggressive but achievable" estimate for critical chain tasks is used to develop the primary characteristic of the critical chain schedule – the buffers. A project buffer, which protects the final project due date from the variability in performance on those tasks is built from the estimates associated with the critical chain tasks. Feeding buffers, which are related to chains of tasks that feed into or merge with the critical chain, are similarly sized and placed to isolate the critical chain from the integration effects of those chains, essentially helping to keep the critical critical (Patrick, 1999a).

Critical Chain Schedules and Risk Assessment/Acceptance

Once developed, assessment of the full schedule, including the contribution of the buffers to project lead-time, provides a clear view into the identified potential of schedule risk for the project. In non-Critical Chain environments, when contingency is included, it is often hidden, either in management reserve, or in internal and external commitments. The common practice of keeping these components off the table hides their true impact and implications. The open and explicit communication of buffers (important as we will see in the discussion of project control) allows a clear assessment of what could happen "in the best of all possible worlds," versus what might happen if individual concerns accumulate to affect project performance.

The ultimate risk of a project is not delivering the promised value in the required time frame. If the schedule results in a lead-time that does not support business needs of the project, the critical chain schedule provides two primary sources for reduction – the critical chain and the project buffer. Assumptions that have been made on key critical activities can be revisited to assess whether additional actions or activities can be added to the project to reduce variability and the size of the project buffer, or whether task handoffs can be restructured to allow more parallel activity and reduce the length of the critical chain. At some point, limits on corrective action are reached, resulting in a buffered schedule that reflects the accepted risk of the project's lead-time and schedule promise.



Critical Chain Schedules and Integration Risk Avoidance/Mitigation

While a lot of emphasis is placed on the project buffer and its protection from critical chain variability, feeding buffers are just as important. They serve to protect project promises from a universal source of risk found in every project that involves parallel activity. Integration risk, i.e., the statistical nature of merging parallel paths, is the primary source of changing critical paths in traditionally managed projects. If a set of parallel paths of activity each have a relatively safe 85% probability of completion by certain point in time, it takes only 4 such paths to turn the chance of an on-time start for the task they integrate into to 52% -- not much more than that of a flip of a coin. When one considers that projects are typically made up of integrations of integrations or integrations, there is little wonder that critical paths change during the life of a projects, and that there is difficulty bringing projects in on time without relying on heroics or hoop-jumping.

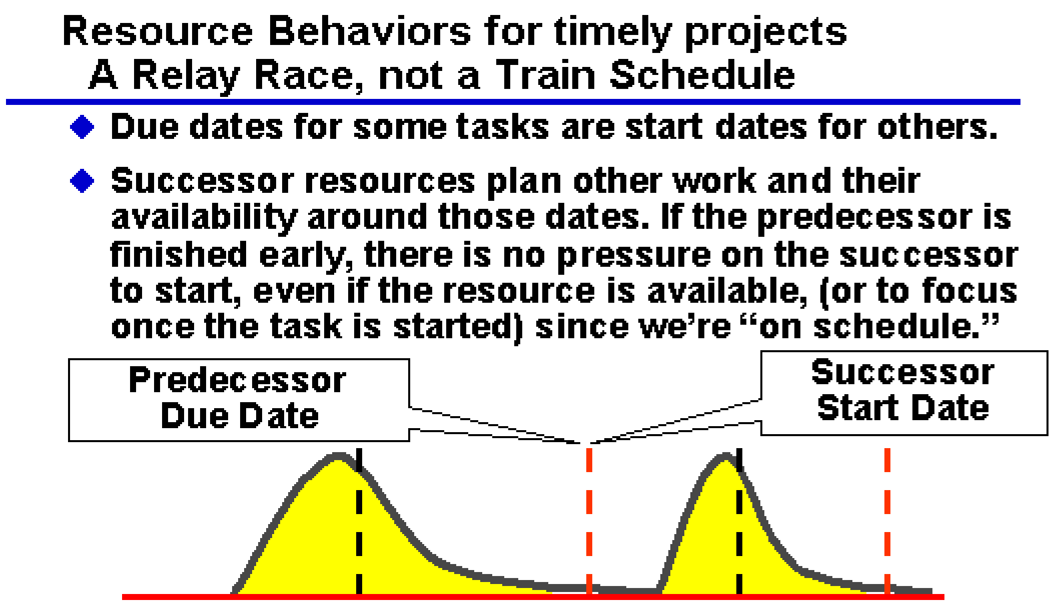
A common tool for assessing this characteristic of risk in traditional critical path project schedules is Monte Carlo simulation, which provides a view of the impact of these integrations on the probability of promised project completion. The critical chain schedule takes these integrations into account up front by explicitly building feeding buffers to deal with the variability in feeding chains (rather than relying on random amounts of slack or float). While Monte Carlo simulations advise on the probability of keeping promises, buffered critical chain schedules are designed to avoid integration risk and keep that probability high.

RESOURCE BEHAVIORS
– MINIMIZING THE EFFECT OF PARKINSON'S LAW

Non-Critical Chain-based projects often rely on safety embedded within tasks and task due dates (milestone schedules) to schedule and control projects. This approach runs the risk of suffering from the impact of common resource behaviors that will minimize the ability to gain time on the schedule. Parkinson's Law – "Work expands to fill the time allowed." – is a resulting reflection of these behaviors. Since ultimately, project performance hinges on appropriate behaviors, the underlying purpose of Critical Chain methodologies is to provide policies and procedures that support desired practices.

A Relay Race, not a Train Schedule

Most projects are managed by carefully watching the calendar, comparing where we are today against some baseline schedule. That schedule typically consists of a series of start and due dates for consecutive tasks, with due dates of predecessors matching start dates of successors. Like a train schedule, if a task arrives at its completion on or before its due date, that portion of the project is considered to be "on track." Successor resources plan other work and their availability around those dates. If the predecessor is finished early, the successor resource may not be available to pick up the handoff. Even if the resource is available, there is commonly little or no urgency for the successor to start (or to focus on it exclusively), since we're "ahead of schedule," and that resource will typically tend to other priorities.



The problem with this common practice is that while it is important for trains to arrive at and depart from their stations (their milestones) at appointed times, project value is more often tied to the absolute speed from beginning to end. The sooner the entire project is completed; the sooner project benefits can be accrued. A more appropriate metaphor to guide projects is a relay race, in which resources are encouraged to pick up the input handoff as soon as it is available, "run with it" in a full, focused, sustainable level of effort, and hand off the output as soon as it is complete.

This behavior is exacerbated in environments where schedules are built upon estimates that are considered commitments by the resources, and therefore contain a substantial amount of localized safety in each task to protect that commitment. If a project is deemed "on track," and a resource realizes that there is chance of completing the work well within the "safe" estimate, the desired sense of urgency is again diminished. As a result, resources are momentarily comfortable sharing their time among several tasks or issues, extending out the time that they would otherwise be able to hand off their output to the next leg of the relay race.

Milestone schedules, like training schedules, become, at best, self-fulfilling prophecies, at least in terms of expectations of speed. They may still (and often do) take longer due to being derailed by Murphy's Law because they have wasted what might have been early finishes which are now not available to offset tasks that take longer than anticipated.

Critical Chain Schedules, Resource Behaviors and Risk Mitigation

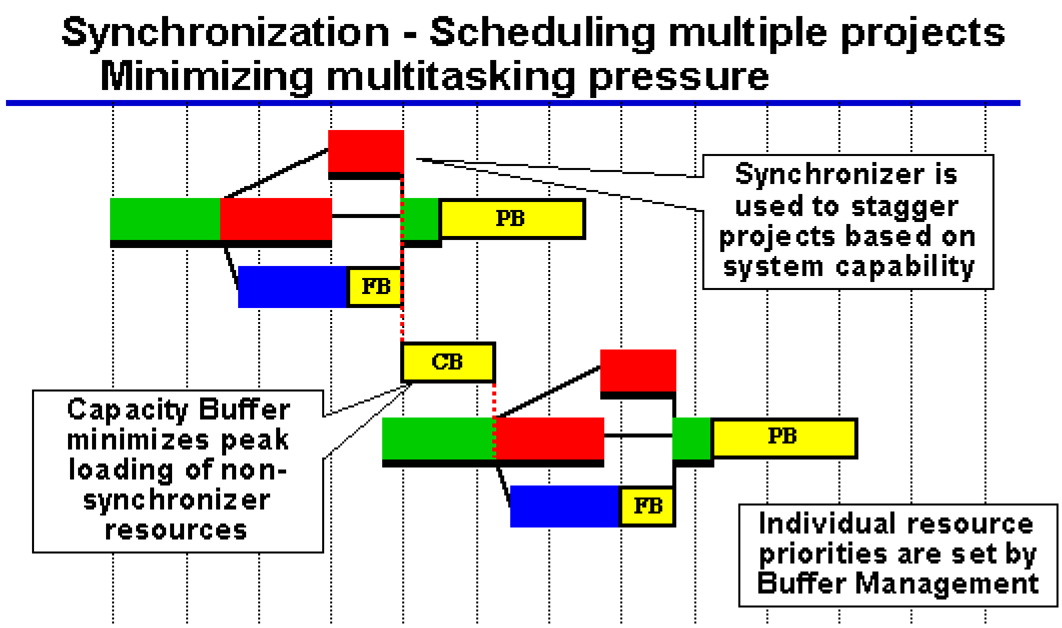
Critical Chain schedules address this question of lost safety in two ways. First, the usual system of task due dates itself is eliminated. The only dates in a critical chain schedule are launch dates for chains of tasks that have no predecessors, and final due dates associated with deliverables that are external to the project and which are protected by project buffers. Start dates of tasks are linked directly to the completion of their predecessors, and communicated through the buffer management project control process, discussed in more detail later in this paper. If you have no due-dates, you have gone a long way in eliminating due-date behaviors and in repealing Parkinson's Law.

Secondly, the safety is moved out of the tasks to the buffers, thereby eliminating the idea of commitment that needs to be protected on one hand or that is good enough on the other. With the underlying assumption that the work of a task will take as long as it takes, no matter what the schedule model assumes, resources are directed to work on tasks without distraction until complete and handoffs are delivered. At least tasks won't be delayed by outside influences. More importantly, management also must support the ability to do so, avoiding unnecessary distractions or conflicting priorities. If resources run their leg of the relay race in an effective and efficient manner, some tasks will take longer than anticipated in the schedule and some will take less. The project is in a position to take full advantage of early finishes. In this way, the cumulative risk associated with due-date behaviors is replaced by the consumption and replenishment of buffers.

SYNCHRONIZATION OF THE PIPELINE
– MINIMIZING RISK OF CROSS-PROJECT IMPACTS

In the previous section, the avoidance of distraction from the task at hand was identified as a critical component of "relay race" resource behaviors. Any time working on something other than the task will extend the time of completion of that task, delay handoffs, and threaten the ability to accrue maximum value from the project.

Most project environments do not have the luxury of being able to focus on only one effort. Most project environments are multi-project environments, where key resources are shared across projects and have to deal with contention for their attention. As a result, while a particular single project may be carefully planned, with effective risk management applied within its borders, it may still be subject to programmatic risk, particularly related to availability of resources that are involved in other, equally important projects.



Synchronization - Scheduling multiple projects

The TOC multi-project solution recognizes that the effectiveness of individual projects can be threatened if the organization tries to push more projects through its pipeline than it is capable of. Scheduling – the actual promising of individual project completions – must take into account any constraining aspects of that pipeline. While the common existence of practices like multi-tasking or due-date behaviors typically prove out to outstrip any actual resource constraint, the possibility of such a constraint is useful as an implementation tool for the multi-project aspect of the approach (Patrick, 1999b).

The process of synchronizing project launches to the ability of a commonly, heavily used resource to deal with those projects helps to minimize pressures to multi-task from the start. This process starts with a review of projects in the portfolio for the identification of potential candidates for the choice of a gating/synchronizing resource. The choice of one that is commonly used across projects and relatively heavily used compared to other resources will suffice.

The second step is to prioritize the current projects, in terms of criticality of current commitments, value to the organization, and use of the synchronizing resource. To the extent that there is no easy consensus of strategic priority for existing projects (a rare occurrence), basic TOC principles of throughput per constraint unit and throughput dollar days can be applied to this effort. The objective of this prioritization is to provide an order in which projects are scheduled through the synchronizing resource.

Once these priorities, procedures and processes are in place, individual project schedules can be developed and put into the calendar through the synchronizer schedule. If chosen correctly, and further protected with capacity buffers, the careful scheduling of this commonly, heavily used resource will result in a set of schedules in which any concerns about contention for other resources will be with the ability of buffer management to provide direction.

Synchronization and Risk Avoidance

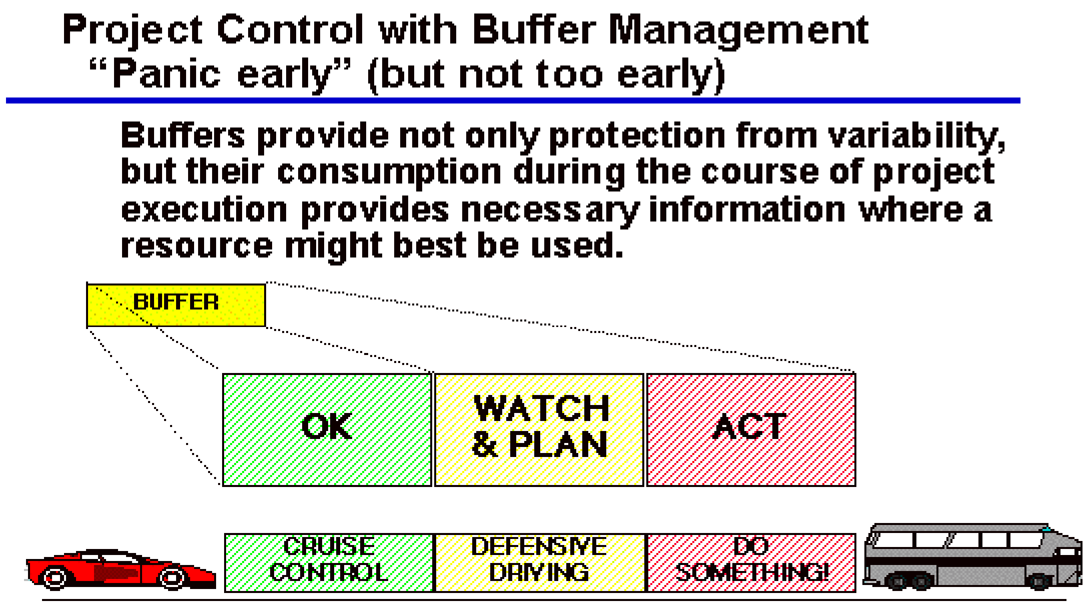
When you consider the duration-multiplying effect of multi-tasking, it should be clear that multi-project risks of cross-project interference could dwarf risks associated with the individual projects. If project value is time-sensitive, the delays suffered by projects due to resource time slicing across projects can be very expensive indeed.

The replacement of systemic pressure to multi-task with synchronization, combined with the management of resources for “relay race” behaviors will go a long way to reduce programmatic risk and to speed project completions across the portfolio. The combination of the two will help avoid having to deal with hard-to-predict cross-project risks. In addition, the required careful consideration of the makeup of the pipeline and the active management of the critical resources identified and used as the synchronization mechanism will aid in understanding potential weak links for future improvement.

Most importantly, if combined with effective and supporting processes for planning, scheduling, and control, synchronization of a project portfolio serves to minimize the overall risks to optimum bottom line performance of the organization that owns the projects and their outcomes.

PROJECT AND RISK RESPONSE CONTROL – CLARITY OF PRIORITIES AND CORRECTIVE ACTION

At several points in this paper, the need for and benefit of effective project control has been highlighted. Planning, scheduling and synchronization are all processes that will create a model of expectation for the project organization. But that model needs to be managed once it comes into contact with reality. Appropriate resource behaviors, especially the required focus on the most important task at hand, require the occasional guidance to clarify priorities in a shifting situation. And if the critical chain scheduling process is used, something needs to be used to replace task due-dates to assess the health of project promises.

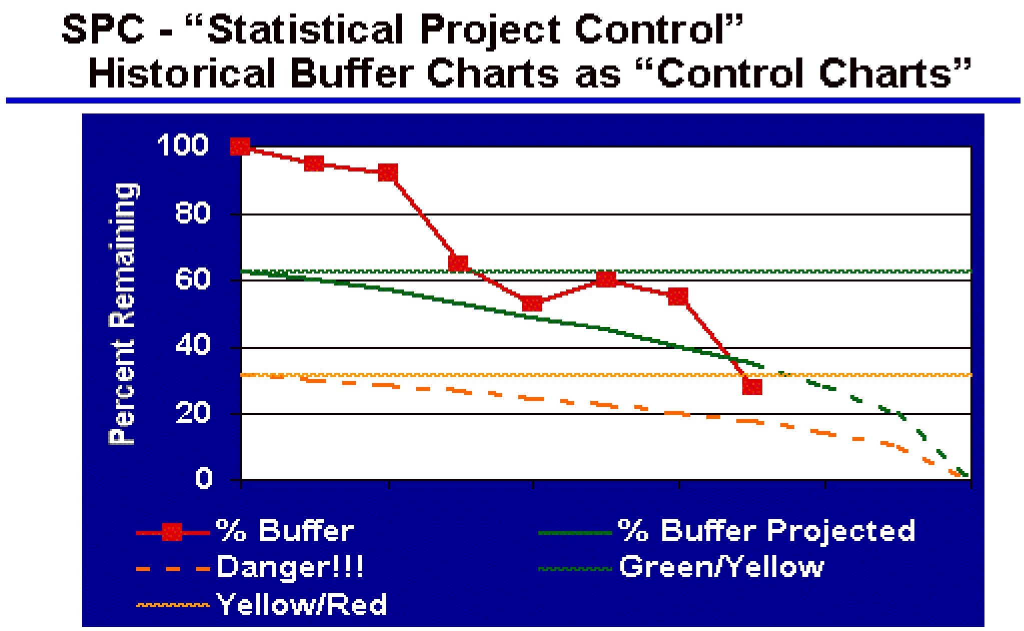


Project Control with Buffer Management

The buffers introduced in the Critical Chain scheduling methodology do not only serve to protect project promises in a static manner. They also provide an ongoing view of the health of the project as reality impacts the expected model that is the original schedule. As tasks take longer than the schedule anticipates, buffers are consumed. As they take less time, those buffers are replenished. Awareness of project buffer consumption relative to the completion of the critical chain (and to the expected variability of the remaining work on the chain) provides an important forward-looking focal point for managing project execution.

A number of straightforward ways of assessing buffer consumption make it clear to everyone involved when and where corrective actions need to be taken. Effective Buffer Management is a critical factor in successful implementations of Critical Chain-based project management systems.

Buffer Management typically involves a combination of real-time access to buffer condition and periodic "buffer management meetings." Real-time, daily updates of project and buffer status are feasible in a Critical Chain environment due to the simple data needed to update active tasks. That data requires only one number at the end of each day – a current estimate of time to complete the task at hand. Immediate issues can be quickly identified through this process.



Periodic multi-project buffer management meetings, typically involving project owners, project managers, and resource managers, start with buffer status of the portfolio's projects. Those with buffers "in the green" require little if any discussion. Those "in the yellow" or "in the red" are rightfully the focus of the meeting, with project managers highlighting identified opportunities and actions for buffer recovery (Patrick, 1999a). These meetings are also useful for supporting regular, forward-looking risk management as well, again with an eye to current buffer condition and to its ability to absorb the impact of identified risks.

Buffer Management and Risk Identification

Consistent buffer management is a major contributor to the establishment of a risk management culture in a particular project environment. Risks and their positive flip side – opportunities – are, by definition, potential future occurrences that require a forward-looking approach to support their identification. The everyday process of developing an estimate-to-complete task status keeps short- and immediate-term risks in the forefront of the mind of the reporting resources. In addition, the elimination of task estimates as commitments and the related transfer of safety to the buffer should support a greater willingness to raise concerns, if the buffer is there to absorb them and they are not expected to have to have an immediate solution to protect their personal performance.

Buffer management also provides a clear view of the cumulative risk effects of project performance. Buffer consumption at any point in time is the result of all previous work, which can eat away at the buffer quietly but insidiously as the project progresses. If buffer consumption is tracked against the amount of chain completed, or alternatively if buffer remaining is tracked against the amount of buffer required to protect what remains of the chain, trends of diminishing buffer condition or the crossing of pre-determined thresholds will serve to identify indications of risk for the project as a whole.

Buffer Management, Risk Assessment and Response Control

Once a possible risk is identified via its impact on buffers, assessment of whether is deserves further attention is required. There are two mistakes that can be made in dealing with identified risks – not acting on them if action is indicated and acting on them if they don't really matter. Project managers are probably sufficiently paranoid so that the risk of not acting is relatively remote. However, that same paranoia can sometimes drive analyses and actions that are not really necessary. And those unnecessary actions will only serve to distract resources and managers from getting on with the necessary work.

Buffer charts, tracking buffers condition against chain completion or buffer required for remaining work, can be utilized in a way that is not unlike the way control charts are used in statistical process control for production environments. For an identified risk, a "what-if" analysis can be easily performed, resulting in a view of the schedule or budget buffer after its run-in with the concern. If sufficient buffer remains for protection of the promise from the variation anticipated for the remaining work, then it is not worth the time and attention necessary to develop corrective actions. In this way, buffer management as risk response control has, embedded within it analysis useful for assessment of individual risks as well.

Buffer Management and Risk Mitigation

The quality of actions taken to avoid or mitigate identified risks is highly dependent on the quality of thinking that goes into their design. The quality of thinking applied to a situation is highly dependent on the environment in which it takes place. With buffer management as the primary project control mechanism, consideration of corrective action takes place when buffer status leaves what is commonly referred to as the "green zone" and crosses into the "yellow zone," or when trends of accelerating buffer consumption are detected. These assessment triggers occur when there is still considerable buffer, and therefore allow the necessary thinking to take place in an environment that is not one of "panic."

If, on the other hand, it does threaten to move the buffer "into the red," then the required mitigation needed to protect the project promise – in terms of buffer reclamation necessary to bring it back to "the green" – provides guidance on the magnitude of the required corrective action.

THE THEORY OF CONSTRAINTS
– MORE THAN CRITICAL CHAIN PROJECT MANAGEMENT

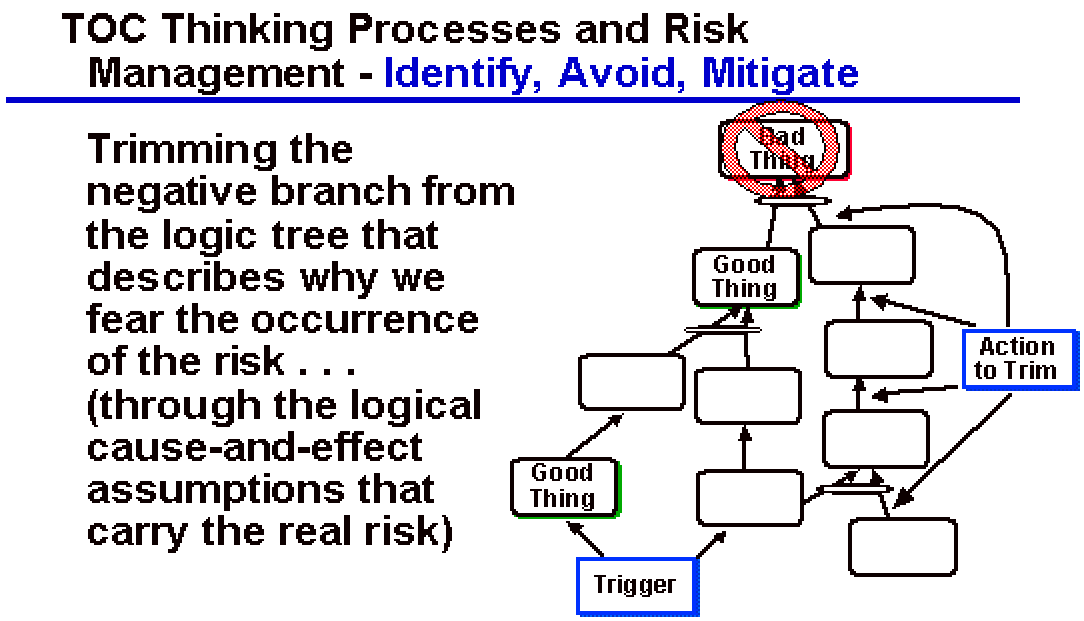
Hopefully, this paper has, at this point, made a case for the TOC approach to project management – Critical Chain Scheduling, Buffer Management, and Synchronized Multi-Project Management – as a coherent approach to project management that supports the basic processes associated with risk management for the protection of project value. But TOC is more than this approach to project management. Actually, the project management solution was derived by applying a set of logical thinking tools – tools that reflect the "hard science" origins of this management philosophy – to the problems commonly faced in the realm of project management (Goldratt, 1994).

Some of the "TOC Thinking Processes" are also applicable in stand-alone situations and should be in the toolkit of a project/risk manager. The first is the Negative Branch Reservation (NBR), a tool that helps to define the concern about a risk in a way that lays out its source in the current situation and the logical cause-and-effect steps that will lead to it. The second is known in various circles as the Evaporating Cloud or Conflict Diagram and is a graphical description of a dilemma or conflict used to raise assumptions about the situation that have within them the potential for a solution.

NBRs and Risk Identification/Assessment/Response Development

"If you define a problem (a risk) well, you probably have it half solved."

There is a lot of truth in that old saw, and it applies to the value of risk identification directly. An identified risk cannot only be the expression of fear of a particular outcome. For identification of a risk to be useful in its assessment and in developing action to deal with it, it needs to include sufficient clarity against which appropriate thought can be applied.



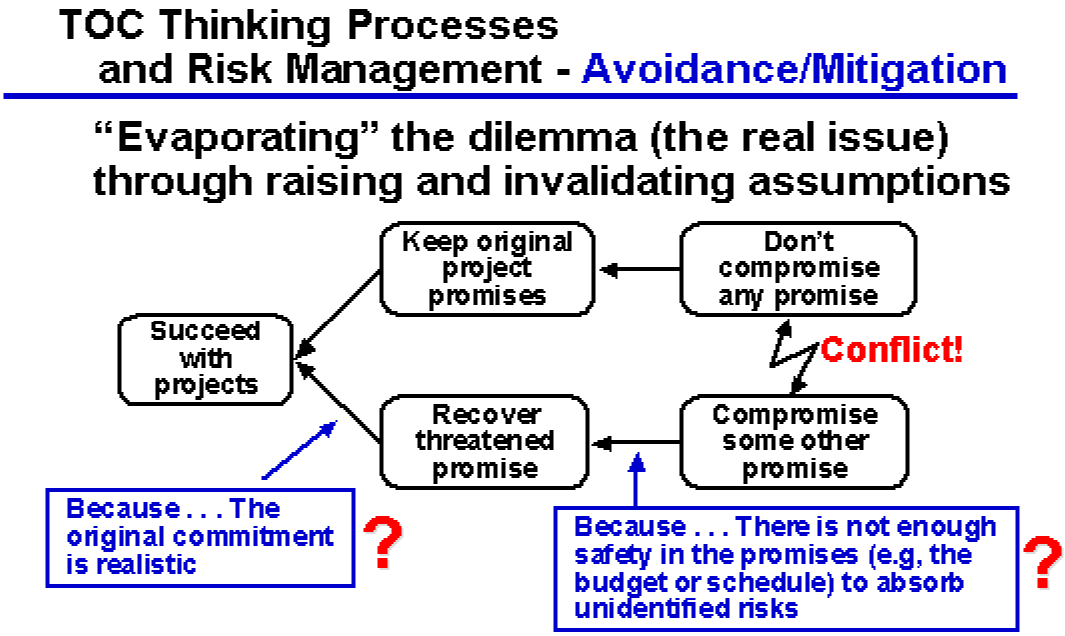
Risk and related issues can benefit from the NBR and the Cloud for developing this clarity. If the risk is related to a concern, a reservation, or seems to be expressed with “Yes, but…” the question may still remain as to if or how that risk may actually occur. The NBR will lay out the path from the starting situation to the concern with as fine a sense of logical cause and effect as is needed to understand it and deal with it. Built with repeated links of if-then-because, the NBR clarifies how the risk will come to pass. This is particularly useful for dealing with questions of technical risk when there is a question of sufficiency of the solution or of undesirable side effects.

If one thinks of the NBR as a “negative branch” on the cause-and-effect “logic tree” leading to the desired objectives of a project, then “trimming” that branch is the path to its avoidance or mitigation. Every logical if-then-because connection along the way has a range of assumptions associated with it. The process for trimming NBRs is one of continuing to ask why certain links must occur, identifying previously unverbalized assumptions that can then be questioned and replaced by new actions, which will then break the logical link. If the cause-and-effect chain leading to the risk is broken, the risk will not occur, at least through that path.

Evaporating Clouds and Risk Identification/Assessment/Response Development

The second need for clarity arises when a situation seems to put one “between a rock and a hard place.” In a case such as this, there are usually certain necessary conditions associated with the objective of the project, which lead to a dilemma or a conflict between the courses of action perceived to support the two conditions. The Evaporating Cloud utilizes necessity logic – “in order to, we must…” – to define the necessary conditions (the needs) of the objective and the perceived requirements to make them happen. Once clearly verbalized, the conflict or dilemma can be confirmed as an issue that requires solution.

Once clearly identified and understood, the process to “evaporate” the dilemma follows a similar path to that of trimming the negative branch, inasmuch as it provides a context for raising and invalidating unverbalized assumptions. Between each “in order to, we must” connection of the cloud, the insertion of “because…” statements is the common approach to bringing these assumptions to the surface. At some point, certain assumptions become suspect, and then provide a way out of the dilemma as they suggest possible other ways of satisfying the real necessary conditions. The real risk is not the conflict that freezes action, but the unverbalized assumption that perpetuates that conflict (Patrick, 2001).



SUMMARY – A FORWARD-LOOKING APPROACH TO FUTURE RISKS

Critical Chain-based Project Management and the Theory of Constraints Thinking Processes provide a range of tools and processes to support Risk Management and the protection of project value. A common thread them is a forward-looking approach to the management of projects. Planning with Network Building looks forward to the objectives of the project before considering the path of activities to get there. The Critical Chain Schedule looks forward to the final project deliverables without being distracted by intermediate task due dates that only serve to sub-optimize schedule performance. “Relay race” resource behaviors look forward with fine focus on the making timely handoffs with quality. Synchronization looks forward to the capabilities of the pipeline. And Buffer Management eschews percent complete or earned value of completed work as water over the dam, and instead looks forward to the work remaining, and its variation and risks.

Summary - Risk Processes and related Critical Chain PM Processes

Risk Processes	CC Planning	CC Execution
Identification	▪ Network Building	▪ ETC Reporting ▪ Negative Branches
Assessment	▪ 2-Point Duration and Iteration Estimates	▪ Buffer Management
Response Development ▪ Avoidance ▪ Mitigation ▪ Acceptance	▪ Network Building ▪ Buffer Sizing ▪ Buffer Concept	▪ Negative Branches ▪ Conflict Diagram ▪ Network Rebuilding ▪ Buffer Management
Response Control		▪ Buffer Management

Management of uncertainty and risk in an effort to deliver promised project value with certainty is what project management is all about, and risk and uncertainty lie in the future. Critical Chain Scheduling and Buffer Management is not only a technique for the development and tracking of project schedules. It is a coherent and comprehensive approach to project management that encompasses and effects other processes and practices associated with project management as well. Most importantly, its implications for looking forward and taking appropriate actions for accepting, avoiding, and mitigating risk are significant and beneficial.

REFERENCES

Duncan, William R., ed., (1996) A Guide to the Project Management Body of Knowledge, Project Management Institute, Upper Darby, PA.

Goldratt, Eliyahu M., (1997) Critical Chain, North River Press, Great Barrington, MA.

Goldratt, Eliyahu M., (1994) It's Not Luck, North River Press, Great Barrington, MA.

Jacob, Dee, (1998), Introduction to Project Management the TOC Way—A Workshop, The A.Y.Goldratt Institute, New Haven, CT.

Patrick, Francis S., (1999a) Getting Out from Between Parkinson's Rock and Murphy's Hard Place, PM Network, 13(4), 57-62.

Patrick, Francis S., (1999b) Program Management – Turning Many Projects into Few Priorities with TOC, Proceedings: PMI International Symposium.

Patrick, Francis S., (2001) Using Resistance to Change (and the TOC Thinking Processes) to Improve Improvements, Proceedings: IIE Solutions Conference, Institute of Industrial Engineers, Norcross, GA.

Pritchard, Carl L., ed., (1997) Risk Management Concepts and Guidance, ESI International, Arlington, VA.

About the Author

Francis S. "Frank" Patrick (fpatrick@focusedperformance.com) is founder and principal consultant of Focused Performance and an Associate in the A.Y.Goldratt Institute network of consultants and educators practicing in the realm of the organizational effectiveness. As such, he has been involved in assisting a wide range of organizations to achieve breakthrough solutions for significant results through the application of the Theory of Constraints. Focusing most of his efforts for the past several years in the area of Critical Chain-based project management in product development, IS, and engineering design, Frank has become a recognized advisor, writer, and presenter on the topic.