CUTTER IT JOURNAL
The Journal of Information Technology Management

CUTTER CONSORTIUM
ACCESS TO THE EXPERTS

March 2003 Vol. 16, No. 3

Critical Chain Project Management: Coming to a Radar Screen Near You!

Project Management Revolution

Critical chain project management promises to revolutionize the project and resource management practices in every corner of the companies that do a lot of project work.

Unknown Fad?

Most project management professionals don't even know what critical chain project management is. If it's so powerful, why doesn't everyone know about it?

"A question people commonly ask about CCPM is, 'If critical chain is so superior, why isn't everyone doing it?'"

— Bill Lynch, Guest Editor

opinion

Opening Statement

by Bill Lynch, Guest Editor

WHERE ARE WE GOING WITH CRITICAL CHAIN?

This issue of *Cutter IT Journal* is devoted to examining critical chain project management (CCPM). This innovation in project management is — as our first article will attest — still unknown in many circles, but it has already changed the experience and results of project teams in a wide variety of industries and companies.

For the past 15 years of my life, I have focused on learning, applying, and institutionalizing breakthrough business processes. While learning and becoming certified in more three-letter acronyms than I care to disclose, I've had the opportunity to witness amazing results and improvements. I've also witnessed mammoth change programs fail, despite many millions of dollars in funding and more resources than the organizations in question knew what to do with.

The discussions that accompany the introduction of new ideas and techniques have always puzzled me somewhat. While much is said, the amount of substantive discussion tends to be minimal. Instead, people share rumors, impressions, and off-the-cuff remarks they've heard from friends of friends. At times, the process is reminiscent of urban legends. Amid some stories of success, disparaging comments abound: "*Nothing new … not practical … not needed … I found a company that's trying to implement this for others, but they're not making any money … I've heard about a company that tried this, and it didn't work.*"

A question people commonly ask about CCPM is, "If critical chain is so superior, why isn't everyone doing it?" This is like the joke about the two economists walking along and spotting something on the sidewalk. "Isn't that a $100 bill?" asks the one. "It couldn't be," says the other. "If it were, someone would have already picked it up."

Behind this simple rebuttal lie more serious reservations facing any new approach that promises great results and requires initial investment of time, money, and learning to achieve. There will always be a healthy level of skepticism toward such solutions. We shouldn't even be surprised when we find there are companies that absolutely believe that the new approach is far superior but still don't move ahead, for reasons we don't fully understand.

Critical chain project management promises to change product development. This might be insignificant if we didn't live in a time when organizations are expected to continually reinvent products and processes just to compete. Projects have become the means of focusing knowledge workers on the results that drive sales, profits, share prices, and salaries. CCPM matters because it holds the keys to changing the experience and results of project teams. We haven't offered an article on why projects matter. You know that they do, and that's probably why you've read this far.

The two preceding paragraphs suggest that resistance to or acceptance of a new process depends purely upon reason. One difference between product innovation (for example, the Segway) and a process innovation is the matter of adoption. An innovator can prove that, say, a light bulb works without having to explain the underlying concepts and principles to customers of the bulb. By contrast, an innovation in how we do something requires the people who adopt it to understand and accept it before "the light bulb goes on." It's not enough for the innovator to act on his or her convictions. The people adopting the process are the ones who must act. The adoption of new processes requires more than reason — it requires a gut-level conviction that the approach deserves time and attention.

I'd like the readers of this special issue to take a long-term change management perspective and to read the accompanying articles with an open mind. At some stage in development, almost all mainstream management practices have weathered resistance from defenders of conventional wisdom. In the late 1980s, just-in-time concepts were in their infancy in the US. The same could be said of lean manufacturing in the mid-1990s, kaizen in the 1980s, and manufacturing resource planning in the 1960s — all had their beginnings as relatively unknown (or misunderstood) concepts. Yet over a relatively short period, the concepts are now widely embraced.

As with any major new approach, there are many years of implementation and institutionalization left to do in large mainstream organizations before CCPM will be seen as a common best practice. This issue contains a series of articles that span the continuum of CCPM experience — from a PM practitioner who was unfamiliar with critical chain before seeing the call for papers to a basic introduction of critical chain concepts, to a success story from a very large organization where critical chain is being institutionalized as the required PM methodology.

Our first article offers the perspective of a consulting professional who has spent decades working in various systems and project management roles. David Higgins' first notion that there was something out there called critical chain project management was when he was approached as a potential author for this issue. He provides convincing evidence that critical chain is not well known within some circles of the PM community. One of his sources contends that CCPM's advance has stalled since the economic downturn because its primary advantage is speed, and companies are focused on cost cutting in these hard economic times. While the article presents what is probably a fairly typical reaction when a long-time PM practitioner researches "critical chain" among his peers, it fails to uncover stories outlining critical chain successes.

In contrast, Richard Zultner, a Senior Consultant with Cutter's Agile Project Management Advisory Service, provides a thorough overview of the critical chain approach as it is promoted by enthusiasts. If you don't know a buffer from a buffalo, or if you just need a CCPM refresher, his tutorial on critical chain principles and practices will get you up to speed. Zultner begins his article with some controversial opinions about the effectiveness of "standard" project management. Whether you agree or disagree with Zultner's take on the current state of PM practice, you'll come away with a clear idea of what makes CCPM different.

The third article, written by Doug Brandt of Abbott Diagnostic Division (ADD), details the journey leading to successful implementation of CCPM throughout a multi-billion-dollar organization. He outlines the decision criteria ADD used to choose a vendor,

Cutter IT Journal®

Cutter Business Technology Council: Rob Austin, Christine Davis, Tom DeMarco, Jim Highsmith, Tim Lister, Ken Orr, Ed Yourdon

Editorial Board: Larry L. Constantine, Bill Curtis, Tom DeMarco, Peter Hruschka, Tomoo Matsubara, Navyug Mohnot, Roger Pressman, Howard Rubin, Paul A. Strassmann, Rob Thomsett

Editor Emeritus: Ed Yourdon
Publisher: Karen Fine Coburn
Group Publisher: Bruce Lynch
Managing Editor: Karen Pasley
Production Editor: Linda Mallon
Client Services: Carol Bedrosian

Cutter IT Journal® *(ISSN 1522-7383) is published 12 times a year by Cutter Information LLC, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA (Tel: +1 781 648 8700, or, within North America, +1 800 964 5118; Fax: +1 781 648 1950 or, within North America, +1 800 888 1816; E-mail: citjeditorial@cutter.com; Web site: www.cutter.com).*

Cutter IT Journal® *covers the software scene, with particular emphasis on those events that will impact the careers of IT professionals around the world.*

Subscription rates are US $485 a year in North America, US $585 elsewhere, payable to Cutter Information LLC. Reprints, bulk purchases, past issues, and multiple subscription and site license rates are available on request.

the company's approach to implementing CCPM, and the outcomes of the implementation. Most senior leaders who face the question of how to improve the current state of their project and resource management processes will find something of value in Brandt's article. He clearly points out that successful implementation requires focused planning and attention. He also demonstrates that the benefits of critical chain go far beyond project speed, impacting the bottom line, employee retention, and quality of life.

We follow Brandt's article with an example of how critical chain concepts were blended and used with the principles of agile development to develop a product that almost all of us have heard about — the Segway Human Transporter (HT). Matt Gelbwaks walks us through the iterative approach that enabled Segway to use critical chain scheduling, and he explains how the company bridged what some might consider to be disconnects between agile development and the critical chain methodology.

The last article is by Robert Newbold of ProChain Solutions. Newbold continues to be one of the pioneers in CCPM and is the author of *Project Management in the Fast Lane*. His article addresses the organizational conflicts that arise as management, at all levels, attempts to reconcile the gap between the realities of project execution and senior leadership objectives. For critical chain to be successfully institutionalized in large organizations, those gaps must be consciously bridged as part of the implementation process. Newbold's article makes the case for credible, reality-based decisions.

Any innovative approach faces a series of obstacles. First, it faces a simple lack of awareness, something that Higgins addresses in the first article. People cannot choose what they are unaware of. The second obstacle is understanding. What makes this approach different, and why should we believe that it works? Zultner's article shares the promise and underlying logic of critical chain project management. A third obstacle is sufficient courage or curiosity to apply the new methodology. Brandt, in our third article, shares ADD's journey from initial application to a few projects to rollout and division-wide adoption. A further obstacle is the "Well, it won't work here" attitude. Those of you who think that the methodology might lend itself better to a huge, established enterprise like ADD than to a startup would do well to consider the role critical chain played in transforming the idea of the Segway into reality within just 18 months. Finally, what is required and promised by the new approach has to be compared against the status quo, or what Newbold calls the "Old Game." Perhaps a final obstacle is the simple notion that the methodology matters.

The objective of this issue is to provide you with a solid overview of what CCPM is and the potential it holds for many organizations. As with the introduction of any new approach, you can expect there to be as many perspectives on critical chain as people you can interview. We are extremely interested in hearing your reactions and thoughts.

— Bill Lynch
blynch@prochain.com

next issue

Project Portfolio Management: Blueprint for Efficiency or Formula for Boondoggle?

Guest Editor: Donna Fitzgerald

Project portfolio management (PPM) is, at its simplest, a process that organizations can use to prioritize where they will invest their scarce resources. There have been many successful applications of PPM in IT, yet many organizations struggle in their attempts to implement the process. Critics claim PPM is an expensive analytical technique that doesn't translate well to the world of IT projects. In which camp does PPM's future lie?

In next month's issue, veteran PPM practitioner Donna Fitzgerald examines the use of PPM in several diverse settings and provides food for thought for proponents and detractors alike. Don't miss this chance to discover what PPM could do for your organization.

what the...?

CCPM's Visibility Problem

by David Higgins

When first approached to do an article on critical chain project management (CCPM), I was initially hesitant. I am not exactly unfamiliar with conventional project management, having authored courses and given seminars on the subject for nearly 20 years. I've also actually managed (pun intended) to successfully manage numerous projects for several *Fortune* 500 organizations. I was, however, unfamiliar with the use of the term "critical chain" used in conjunction with "project management." In short, I'd never heard of it.

Now, speaking as a consultant who makes his living working with systems planning and development methodologies, I will be the first to admit that I don't stay up with every single advance in every single area of systems improvement. I'm not sure *anyone* can and still get any real work done. Therefore, I wasn't particularly surprised that I had never heard of critical chain project management. I just assumed it was a new technique that had surfaced while I was off doing other consulting projects. So like any good author, I decided to do a bit of research before I decided whether or not to write the article.

NO STONE UNTURNED

To my surprise, there seemed to be ample information on the Web about critical chain project management going back about five or six years. So it apparently wasn't a particularly *new* technique, after all. And in reading through the literature, it seemed as though CCPM had proven successful in the organizations that had tried it. It didn't look like some academic, ivory-tower philosophy or one suited to only smaller, simpler organizations. Big-name organizations like ITT, Lucent, Seagate, and Harris were having success with it. I therefore assumed that even though I had never heard of the method, I would have no trouble finding a colleague or two who had a firsthand experience to share.

I contacted a few of my acquaintances here in the Kansas City area, that do project management consulting (and actual PM) on a full-time basis, to gather some of their experiences and impressions. What I found *did* come as a surprise. Not only did *I* not know anything about CCPM, *none* of the people I contacted knew anything about the method either.

I assumed I would have no trouble finding a colleague or two who had a firsthand experience to share.

I first contacted my friend Dan McCune, who is president of the Kansas City chapter of the Microsoft Project Users Group-Global (KCMPUG) to see what he knew about CCPM. He wasn't familiar with the term (I think his exact words were "*What* kind of project management?"). I thought that was a bit odd, since in addition to being a great project manager, Dan had just completed his Project Management Professional (PMP) certification through the Project Management Institute. I would have thought that PMP certification would cover any hugely successful project management technique, no matter how recently it had been developed.

My timing in talking with Dan proved fortuitous, however, as it turned out that KCMPUG was meeting later that evening. So I prevailed upon Dan to take an informal poll for me and see if any of the other

members had any experiences they would be willing to share. He called back the next day to let me know that he got a "deer in the headlights look" from the collective membership when he asked them if anyone had used CCPM. No luck there.

Next, I called up my friend and former colleague Monte Beery, who is currently a director of business development with eSecurity Online, an Ernst & Young company. Monte also happens to be a former senior project manager for Black & Veatch (BV), a $2 billion-per-year engineering firm headquartered in the Kansas City area. BV has been in business for more than 75 years and has managed the development of massive construction projects (including numerous nuclear power plants, considered by BV to be the "holy grail" of projects to manage), so I figured that Monte would surely have at least heard of the technique. It turns out he had not, to say nothing of regarding it as any kind of "industry best practice."

No one had ever used the method; no one had ever even *heard* of the method.

Since CCPM seemed to be such a promising technique from what I had read, I thought that it was highly unusual for someone at a Big Four firm to not know much about it. So I contacted a few more acquaintances: some in local and state government, one or two in the aerospace/military industry, and a couple in telecommunications. Same story. Nothing. Nada. Zip. Zilch. No one had ever used the method; no one had ever even *heard* of the method.

YOU MEAN EVERYTHING'S *NOT* UP TO DATE IN KANSAS CITY?

This was starting to get interesting. Now I was not so much interested in finding a firsthand experience as I was in trying to discover why such a seemingly promising technique was not only unused but also unheard of in the PM professional arena around here.

So I called Jim Highsmith, whom I have known since he and I worked with Ken Orr back in the 1980s. Jim, for those of you who don't know, is a Fellow with the Cutter Business Technology Council, directs Cutter Consortium's Agile Project Management Practice, is considered a leader of the agile methodology movement, and has written a couple of very good books on the subject. Jim had written an article on CCPM for Cutter back in October 1998, and he suggested that I contact Tony Rizzo, whom he had interviewed for his article back then. Tony had been with Lucent at the time and had achieved some rather impressive results while using CCPM there. He left Lucent shortly thereafter to form the Product Development Institute. (www.pdinstitute.com), which specializes in CCPM consulting and training.

My call to Tony was initially to see if he had any references that I might contact in the Kansas City area. He said he didn't. I asked if he knew of any reason why critical chain project management wasn't being used here in the middle of the country, and his answer pretty much solved the mystery for me.

PROJECT MANAGEMENT'S IRON TRIANGLE

Before I let you in on what Tony told me, I would like to digress a moment to give you a little background. Back in the 1980s, I worked with Council Fellow Ken Orr and many others in developing the Data Structured Systems Development (DSSD) methodology — perhaps more widely known as the Warnier/Orr approach. My job back then was to research various topics that would extend the methodology and to develop training seminars for those areas of the methodology that we enhanced. One particular seminar we developed was for project planning and project management. In the course of developing this extension to the methodology, we got to study a lot of different project management methods that were in use back then and listen to what one or two PM gurus had to say on the subject.

One of the project management gurus was Larry Putnam, who is,

by the way, still around at QSM (www.qsm.com). Larry had been working since 1978 on his Software Lifecycle Management (SLIM) method and had developed some very interesting software for resource loading and scheduling. (This was way, way before Microsoft Project, mind you.) He attended and spoke at several of our DSSD user conferences in the early 1980s. During one of his talks, he presented one supremely elegant chart that I've used ever since to explain simply and succinctly the fundamentals of project management. By using just three lines and three words, his chart cut to the essence of all PM methods.

Larry's illustration explained that there were only three variables in project management: time, resources, and scope. It looked something like the chart shown in Figure 1.

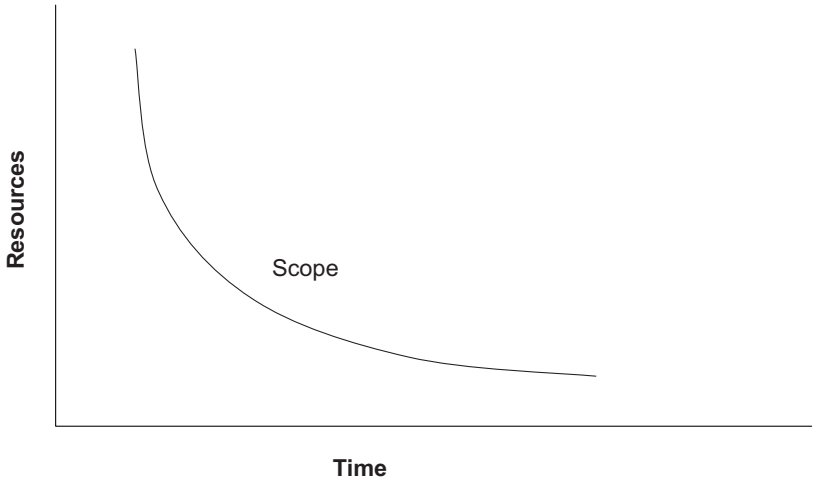

Figure 1 — The three project management variables: time, resources, and scope.

In the figure, resources are shown on the vertical axis, time is shown on the horizontal axis, and scope is shown as a curve illustrating the relationship of time and scope on a given set of project requirements. By adding more resources, the requirements can be implemented sooner; conversely, decreasing the amount of resources will increase the amount of time it will take to complete the project. Increasing or decreasing the scope moves the curve either upward or downward, as shown in Figure 2.

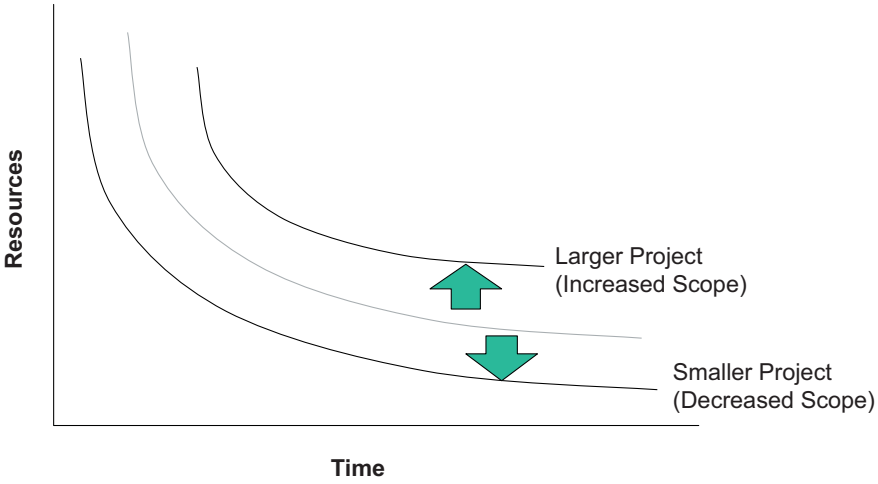

Figure 2 — Increasing and decreasing scope.

I've always found Larry's chart very useful in explaining the essence of project management. As long as at least one of the three variables of time, resources, and scope is not fixed, the others can be manipulated to complete the project successfully. If time and scope are fixed, you can increase resources (more people, more skill, more overtime, better equipment, etc.). If resources and scope are fixed, you can lengthen the time to delivery (see Figure 3). If time

and resources are fixed, you can vary the scope by subtracting project requirements (see Figure 4).

This simple chart provides an elegant illustration of why projects often fail:

- The requirements increase via "scope creep," but the due date and staffing level don't change.
- The due date is moved up, but the resources aren't increased.
- The resources are decreased, but the due date isn't extended.

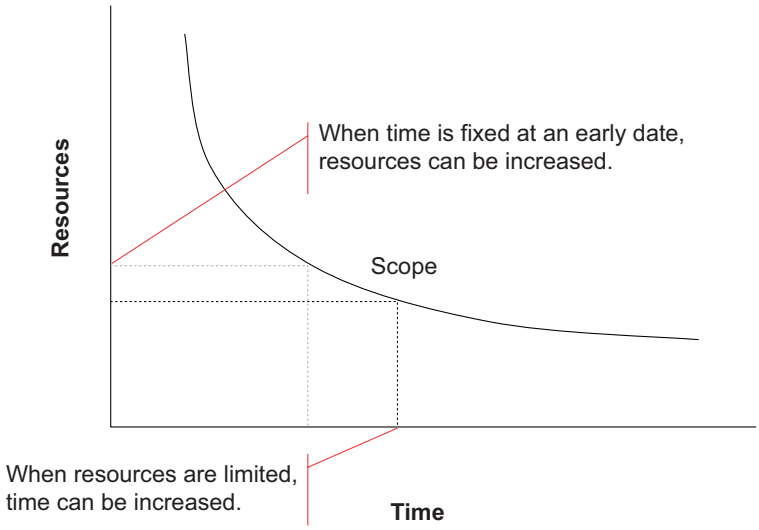

Figure 3 — Making project tradeoffs.

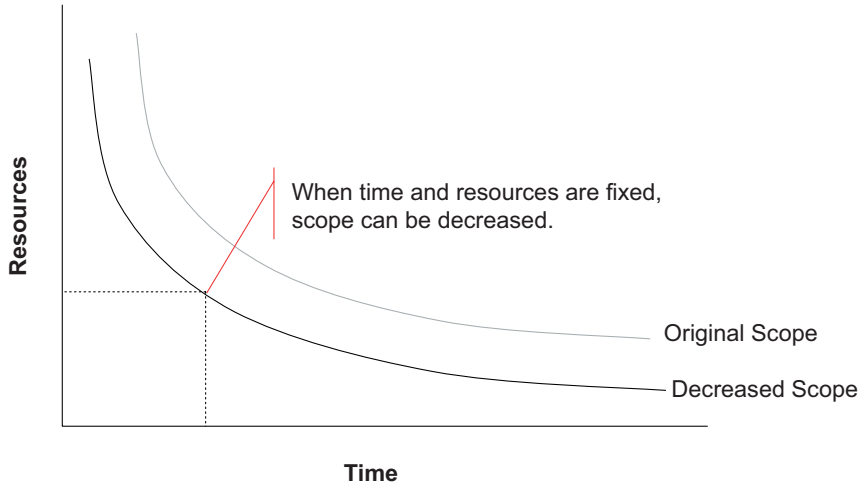

Figure 4 — Time and resource constraints lead to a reduction in scope.

It also illustrates three fundamental truths. The first is this: no matter how many resources you add to a project, there is an absolute limit to how quickly the project can get done. (Even with infinite resources, you can't get the project done in zero time — which is one of the problems I have, by the way, with some of the arguments proposed by Eliyahu Goldratt's theory of constraints.) The second truth is that if you take away enough resources, the project will never get done. The third fundamental truth is one that most executives and many project managers conveniently ignore: when all three variables are fixed by management edict, there is no project management to be done. The project will succeed only if the intersection of the three variables happens to land on the intended scope curve (which it somehow never does). In one of my favorite Dilbert cartoons, Wally sums it up best when he says, "Of all my projects, I like the doomed ones best."

IT'S THE ECONOMY, STUPID

So what does this history lesson and Project Management 101 lecture have to do with Tony Rizzo's response as to why no one in this area uses CCPM? Tony told me that the market for CCPM dried up virtually overnight in April 2000 — and the answer to the riddle suddenly clicked into place: the economy.

You'll recall that April 2000 was about the time when everyone finally realized that the tech bubble

of the 1990s had burst and that the longest bull market in history was indeed turning into a bear. The adoption of CCPM, which had begun promisingly enough two or three years earlier, ground to an abrupt halt. Since CCPM is primarily a technique to help organizations shorten development time, interest in the method quickly evaporated when corporate emphasis suddenly shifted from decreasing time to market to cutting costs. Not only did the marketplace stop spending so much on consulting and on implementing new concepts, it completely lost interest in a method that was principally about saving time, not dollars.

Hence the brief digression back to Larry Putnam's time/resources/scope chart. Rather than focus on cutting time, organizations in today's economy have largely concentrated on reducing resources and cutting back on scope (and in the process, have usually *extended* the project duration). Critical chain project management's scheduling philosophy was never about cutting costs; it was about cutting actual time to delivery and developing schedules that didn't slip. Unfortunately, CCPM's chief selling point of "reducing time" involves the one project variable that CFOs are least interested in decreasing right now.

Unfortunately, CCPM's chief selling point of "reducing time" involves the one project variable that CFOs are least interested in decreasing right now.

Thus the mystery of why no one around here is using CCPM isn't such a mystery after all. Interest in the method isn't just lacking around *this* area; it seems to be lacking just about everywhere these days. And it will likely stay that way until the economy takes a turn for the better.

So what then? Will CCPM take off once the economy turns around, or will it remain a largely unheard-of technique? Time will tell, of course, but I suspect many of the concepts proposed by CCPM — particularly its emphasis on identifying and minimizing the impact of constraints — will ultimately be rediscovered and incorporated into mainstream project management methods.

David Higgins has been an evangelist for high-quality systems development methods since 1975. Together with Cutter Business Technology Council Fellow Ken Orr and the late Jean-Dominique Warnier, Mr. Higgins was one of the principal architects of the DSSD (or Warnier/Orr) methodology. He has given hundreds of seminars on a wide variety of topics, including program design and modification, systems and database design, requirements definition, planning, and project management. Mr. Higgins is the author of five books, including Program Design and Construction*, which was translated into over a dozen languages. His book* Data Structured Software Maintenance *remains one of the few to address the practical application of structured concepts to the modification of existing programs.*

Mr. Higgins has periodically managed systems development for divisions of three Kansas City–area Fortune *500 companies. Most recently, he was Knowledge Management Practice Area Manager for BV Solutions Group, a subsidiary of the engineering firm Black & Veatch. As a management consultant, Mr. Higgins has tackled projects ranging from strategic systems planning to project management, enterprise application architecture, data sharing, data warehousing, knowledge management, business continuity planning, and communications planning for public- and private-sector organizations.*

Mr. Higgins can be reached at 21129 Golden Road, Linwood, KS 66052, USA; Tel: +1 913 723 3569; Fax: +1 913 301 3596; E-mail: dave@davehigginsconsulting.com; Web site: www.davehigginsconsulting.com.

the end of "hurry up and wait"

Getting Projects Out of Your System:
A Critical Chain Primer

by Richard E. Zultner

Do your software projects take too long? For most software managers, the answer is clearly, "Yes." Or perhaps the problem is not the length of your projects per se, but too many late projects or too many projects where scope was reduced or quality sacrificed to make the deadline.

In many software organizations, good project management is applied, but despite careful planning, projects are late. When projects are late, blame is assessed. Usually, some blame is placed on the team and some on the project manager. The next time, the team members, and especially the project manager, try to include even more safety in their estimates ("On our last project we didn't have realistic estimates — look how late we were"). They determine the project will take 10 months. Senior managers, being presented with a project plan with more safety (and therefore a longer schedule than for a comparably sized project), reach for the ax and chop the project schedule ("because that is just too long"). They cut it to eight months.

This article was adapted from "Software Critical Chain Project Management: Do Silver Bullets Exist for Schedule Reduction?" Cutter Consortium Business-IT Strategies *Executive Report*, Vol. 3, No. 10, October 2000.

The project team starts the project with a chopped schedule. Team members suspect it is unrealistic, but they struggle valiantly to try to meet it. Despite their best efforts, they finish in nine months, confirming in their minds that they had no chance from the very beginning. The cycle begins anew. Teams and project managers attempt to add more and more time into their project plans (becoming increasingly resourceful at sneaking safety into their schedules), and management responds by wielding the ax with a heavier and heavier hand, chopping more and more time from every project. Everyone loses.

TAKING A SYSTEM VIEW

Enter critical chain project management (CCPM). With critical chain, we consider all the tasks in a project as a *system*. A project isn't finished until all its tasks are completed. So, what is the goal of a project with respect to its schedule? Isn't it to finish all project tasks by the deadline? To accomplish this, critical chain starts with some familiar facts but reaches a new and exciting conclusion.

CCPM has two parts: one for planning and managing an individual project (single project) and one for managing the entire set of projects in an organization (multiproject). I will begin by discussing single-project critical chain and consider multiproject implementations below.

Critical chain starts with some familiar facts but reaches a new and exciting conclusion.

Single-Project Critical Chain

Estimate Versus Commitment

The different tack critical chain takes starts at the level of a task on a project. When someone in your organization is asked, "How long will this task take?" is he or she being asked for an estimate or a commitment? A statistical *estimate* is the time that the task is "expected" to take half of the time. That is, if half the time a task takes less than five days, and half the time it takes more than five days, then the expected time for the task is five days. To be clear, let's call this the 50% likely estimate.

A *commitment* is the time by which we can rely on the task to be completed. That is, just because a task is estimated at five days (has a 50% likelihood of being done in five days) does not mean we can count

on having everything run smoothly so it is *always* done in five days. What level of likelihood do you feel comfortable committing to? For many software professionals, it is 90%. For example, the time in which you would be willing to commit to finishing the task would be 10 days.

All tasks have variation. No task takes exactly the same time when it is repeated. In software development, we have at best only *similar* tasks — so we have high variability in our tasks (see Figure 1).

As represented in Figure 1, there is a 10% likelihood of completing this task within two days, a 50% likelihood of completing this task within five days, and a 90% likelihood of completing this task within 10 days. If you hit your "estimates" nine out of 10 times, then your estimate is not the 50% likely *expected* time, but a 90% likely *commitment* time. This is what most people provide when they are asked for an estimate. They are being asked for a time by which we can count on the work being done (see Figure 2).

As indicated in Figure 2, an estimate you are willing to commit to, given the high variability we face on software development tasks, includes significant safety time. You add significant safety time because (1) you must allow for your actual working conditions, including the many little urgent things that pop up; (2) the task might be harder than it looks when you get into it; (3) stuff happens (Murphy could strike), so you allow yourself some protection; and (4) it is bad to miss your commitments. You add safety so you can deliver when you said you would, regardless. The safety you have included, the difference between a 50% likely expected time and the 90% likely committed time, is a good measure of the variation of the task. As a general rule, most software tasks are estimated at about a 90% likely commitment time, and about half of that duration is safety — protection from risk.

We include safety to ensure that we can be on time with our tasks. But is everyone trying to be on time with every task the best way to deliver the project on time?

In Figure 3, we have a project with three identical tasks with safety included in each task. We are managing variation at the task level. Project managers try to ensure that the project is on time by keeping every task on time.

To use an insurance analogy, think of three homes where the homeowners attempt to self-insure against risk. Acting in isolation, they will be unable to set aside enough to cover themselves against a serious problem (one that takes more than five days to solve). This was the historical situation centuries ago, before the concepts of insurance and pooled risk were developed. Traditional project management still operates this way. So, with three 10-day tasks to be

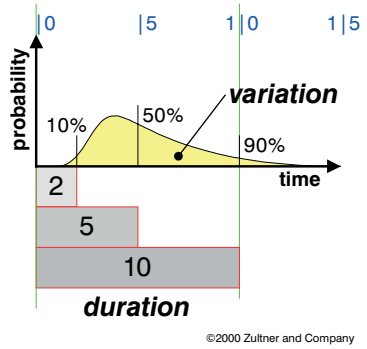

Figure 1 — Estimating the variability of a task.

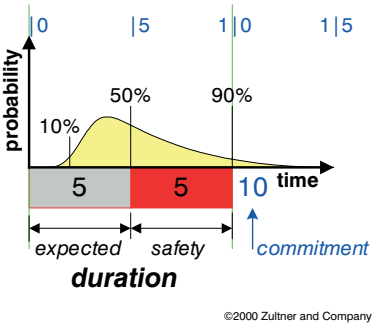

Figure 2 — Developing a commitment time.

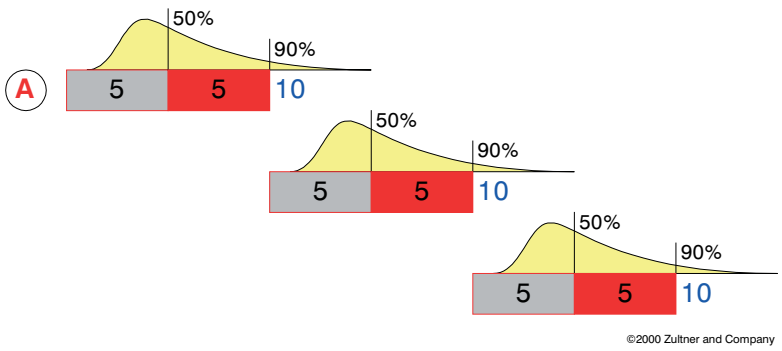

Figure 3 — Project A has three identical tasks, each with a safety margin.

performed and safety required for each, the project will take 30 days to complete.

Each of the three tasks has a 90% likelihood of finishing on time. But the likelihood of the project finishing on time is less than 90% — this is one reason experienced project managers add a *contingency reserve* to the end of the project (if they can, and if it's not cut out by management).

The schedule reduction comes not from magic, or clever compromises, but from the efficiency of pooled safety.

Given that we live in the 21st century, and we know about insurance and pooled risk, how can we improve matters? Instead of inefficiently self-insuring, let's *pool the safety* (see Figure 4). When an insurance company pools the safety of the three homeowners mentioned above, the homeowners pay less to the insurance company than they planned to set aside previously — and they get more protection. How is this possible? Because the insurance company knows that all three homes are unlikely to suffer a serious problem at the same time.

For our software example, the efficiency of pooled safety is such that a shared pool of 8.66 days provides exactly the same protection for the *project* (90% likelihood) as five days did inside each task. But now, a task can withdraw up to 8.66 days from the pool to cover its problems without making the project late. Likewise, three tasks collectively could have problems adding up to 8.66 days without making the project late, because they share the pool: a buffer against risk.

This equivalent protection occurs with a project that is six days shorter — with no other tradeoffs. Instead of a 30-day project, we have a 24-day project (a 15% reduction). Significantly, the gains from efficiency of a shared risk pool *increase* with the size of your projects. This is why we can conservatively say that critical chain single projects are 15%-25% shorter than traditional projects with no tradeoffs. That is, 15% shorter on small projects (like the three-task, 30-day example here) and 25% for large projects (say, 10 tasks and a 100-day duration).

The schedule reduction comes not from magic, or clever compromises, but from the efficiency of pooled safety, a centuries-old insurance concept. We can use the concept of pooled safety to protect against a variety of risks, the first being the risk of the project being late.

The Project Completion Buffer

The pool of safety for a project is called the project completion buffer. As a task finishes, if it takes less than the 50% estimate, the savings are deposited in the project completion buffer. We expect half of the project tasks to do so. If a task takes more than the 50% estimate, days are withdrawn from the project buffer to cover it. We expect half the projects tasks to do so. Now, it doesn't matter if any *particular* task is early or late — as long as the project buffer can absorb the variation, our project will finish on time.

It's important not to chastise people for finishing a task later than the 50% likely expected time. We understand task variation, and we know Murphy exists, so we've planned for risk with a project completion buffer. The only way you will get accurate 50% likely expected task times is if it is OK to exceed them. In fact, if they are not being exceeded half the time, then you do not have 50% likely expected times. This calls for significant behavior change from project managers.

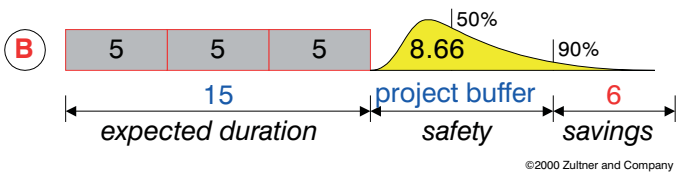

Figure 4 — Pooling the safety.

Note that the project completion buffer provides an explicit means to manage schedule-related risk. Basic risk management is *integrated* into critical chain project management.

Unless we start tasks as soon as possible, we will suffer from variability (bad luck) but never benefit from variability (good luck).

How do the buffers work as risk "shock absorbers" to absorb variation when Murphy strikes? Consider Figure 5. We are continuing with our project of three 10-day tasks, which is planned as a critical chain project with 15 days of expected project duration and nine days of project safety in a project completion buffer. As the project progresses:

1. The first task takes six days instead of the five expected, so one day is withdrawn from the buffer.
2. The second task takes three days instead of the five expected, so two days are added to the buffer.
3. The third task takes 10 days instead of the five expected, so five days are withdrawn from the buffer.

The final result is that we finish early. In this case, the project buffer had been calculated to provide a 90% likelihood of finishing the project, and the result was a 24-day project duration. That means that if this project was done 10 times, we would expect that nine out the 10 times, the project would finish on or before 24 days. But since we reduced our schedule six days from our original 30, we could *invest* some of those savings back into the project completion buffer to raise the likelihood to 95%, 98%, or more. In fact, if we put *all* the savings in the project completion buffer, we would have the same 30-day project we started with, but the likelihood of being late would be *extremely* low.

With CCPM, we can have substantially shorter projects, make late projects virtually extinct, or some of each. You decide, and you control the project to that end.

Acting Like Relay Runners

There are two very important behavior changes required for the buffers to do their job. Often, we don't start tasks as soon as possible. Instead, we wait because (1) we have plenty of time (we included significant safety time in the task estimate); (2) we're busy (we always have a number of urgent little things that need to be done right now); or (3) we're not ready (we haven't arranged to have everything we need to do the work available to us so we can start immediately). Eliyahu Goldratt, the originator of critical chain, refers to this as the "student syndrome."

This behavior can cause the safety included in the task to evaporate even before we start to do the task — we no longer have a 90% likelihood chance of finishing by the committed time.

In Figure 6A, with a 90% likely committed time, we have plenty of time to finish the task, right? So we take care of a few other urgent little things first. In 6B, the first thing that happens is that the safety in the task evaporates because we didn't start the task right away. In 6C, we get into the work, and Murphy strikes. Now we need evenings and weekends just to make our original, "comfortable" 90% likely committed time.

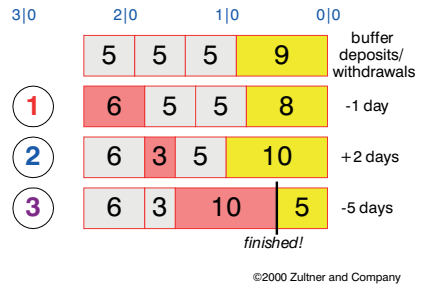

Figure 5 — Buffers absorb shock.

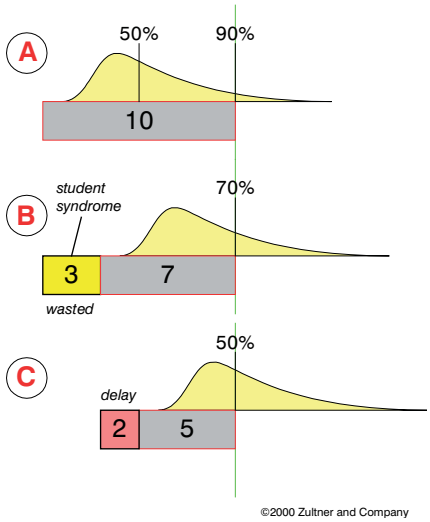

Figure 6 — Task safety evaporates.

Unless we start tasks as soon as possible, we will suffer from variability (bad luck) but never benefit from variability (good luck).

The second behavior change required for the buffers to do their job starts with not finishing tasks when a traditional project schedule says we should. Committed task times include safety, but what if good luck or a brainstorm strikes, and you finish the task early? Will you hand off the task to the next person? Probably not. Why?

- There is always a bit more to do to make it even better (polishing).
- The next person isn't ready to start anyway. ("Why are you bringing me that now? I'm not supposed to start that until Monday. I'm still working on something else.")
- If I finish early, then my estimate might get cut next time (and I might not have the brainstorm as I did this time).

We can explain this as Parkinson's Law (the work expanding to fill the time available) or that we erroneously report when we're done (false finish reporting). We were done earlier but did not hand off the work. So the project gets no gain from our good luck. (If we had bad luck and handed off late, the project would definitely get our loss.)

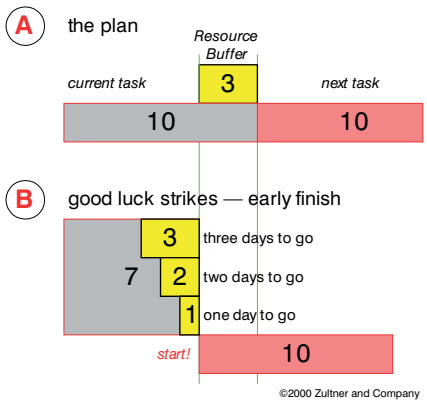

Figure 7 — Using countdowns.

For our people to do tasks faster, we must reduce their workload and lower their utilization.

Thus, the behavior we need on critical chain projects is for people to start work at full speed *immediately* when the preceding task finishes (not when the project schedule predicts they should start). We also need them to work at full speed until the task is finished and then *immediately* hand off the work to the person in the next task. We need our project members to act like relay runners.

How do relay runners behave? At the start of the race, the first runners arrive early, prepare, and *wait* for the start. As soon as the race starts, they run as fast as possible. The next runners in the relay are ready and *waiting*. They are not asked to "make good use of their time" or to "increase their utilization" by working on something else while they are waiting. They wait, so they are able to start running as soon as they get the relay baton. All the runners in the relay race spend most of their time waiting, so that any good luck the previous runners acquire is taken advantage of.

If we are serious about speed, we must understand what is required for people to "go faster." We are *not* asking them to work faster! We are asking them to wait, to rest, and then work at full speed. And then rest again. Our strategy is to remove all the obstacles that cause them delay and to recognize that relay runners don't run at full speed all day. They are not "fully utilized" because full utilization of resources is inefficient for speed. For our people to do tasks faster, we must reduce their workload and lower their utilization. This is not a sacrifice for the organization, it is the means for substantially greater project throughput.

Using Countdowns

How can we make it easier for people to be ready to immediately start their assigned tasks when they are triggered by task completions, not by a traditional project schedule? When a task is approaching completion, we need to alert the person doing the next task to get ready. We have to give him a countdown, or advance warning, so he can start as soon as we're ready to hand off (see Figure 7).

Ask the person how much warning he will need so he can immediately start any assigned task. That warning time is the resource buffer. In Figure 7A, the resource buffer is set at three days. This countdown will give him time to (1) finish or pause any tasks he is working on currently; (2) arrange for whatever he needs to perform the assigned task;

and (3) be ready to start the assigned task immediately.

In Figure 7B, the person doing the first task has been working for four days and thinks she only needs three more days to finish. She starts the countdown, "Three days to go," and the next person begins to get ready. After another day, it looks like she'll be done in two more days. "Two days to go." Then, "One day to go." And finally, "Here it is, go!" The next person, with three days of warning, is fully prepared and starts work at full speed, immediately. (Note: a resource buffer appears in the project plan, but because it is an information buffer, it consumes no time on the project.)

The schedule reduction benefit of single-project CCPM depends on the change to relay-runner behavior by the project team members. Management must employ all available means to support the change and to maintain it.

Multiproject Critical Chain

Traditionally, project management has not offered much guidance for those tasked with managing a large set of projects — beyond treating them as "sub" projects to an organization-wide "super" project. This approach assumes that there is no difference between managing multiple projects and subprojects.

With critical chain, we consider the entire set of projects in the organization as a *system*. Specifically, all the projects that you do this year will come out of the *system* that is your organization. These projects don't deliver any benefit to customers until they are delivered. So, what is the goal of your organization this year? Isn't it to maximize your throughput of projects? To deliver as many projects, as fast as possible, to your customers? To do so would maximize the benefits you deliver.

If our goal is to deliver as many projects as possible to our customers, then what is the worst thing we can do as a manager of multiple projects? In the theory of constraints, many aspects of managing multiple projects have been studied and simulated. Repeatedly, in all the analytic studies and all the simulations, one practice has emerged as the worst and most devastating project practice — a practice that is not just bad project management, but evil project management — bad multitasking of resources. Unfortunately, it is *very* common.

Multitasking is "the practice of assigning one person concurrently to two or more tasks." Multitasking is also known as fractional headcount. One person is assigned to multiple projects simultaneously (or multiple tasks within a project simultaneously). Does your organization do this?

Even if you have "dedicated" project teams, you may still have multitasked resources. If your people are assigned to do multiple tasks concurrently within the project they are "dedicated" to and are expected to show progress on those tasks, then they are multitasked. (If this is your situation, when it mentions three projects in the examples below, just translate that to three "threads" of parallel tasks within a single project.)

In all the analytic studies and all the simulations, one practice has emerged as the worst and most devastating project practice — bad multitasking of resources.

We have known for a long time that there is some "set-up" time that happens when people switch their attention from one task to another — especially from a task on one project to a different task on another. They have to re-familiarize themselves with the work before they are at "full speed" on the new task. Such task switching takes time and has some effect on errors, quality, and so on. But the effect is seen to be small, and the gains in resource utilization (and therefore productivity) from multitasking are seen to be large. That's why we multitask — so resources can be fully utilized, right?

We assume that if every individual is busy all the time, he or she is as productive as possible. If everyone is as productive as possible, since the organization is made up of the people in it, then the organization must be as productive as possible. Is this true? Does local productivity really add up to global productivity? No. Not in a system, as we will see.

Critical chain can reduce the elapsed time for an entire set of projects by 15%-25% (and this is a conservative estimate) by eliminating bad multitasking. Let's look at an example to see how devastating the common practice of bad multitasking is to project throughput.

Figure 8 shows three identical projects, with a project manager responsible for each project. Each project consists of 10 two-day tasks (letters a through j). There are three types of resources required:

1. Resource X is an analyst for analysis tasks a-c.
2. Resource Y is a designer for design tasks d-g.
3. Resource Z is a developer for development tasks h-j.

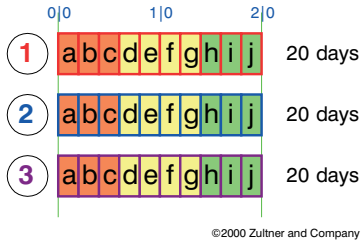

Figure 8 — Ideal case: three projects, dedicated resources.

In this ideal case, since we have three of each type of resource available, we can dedicate one of each resource type to each project. No project will depend on another, and all three projects will be done in 20 days. All the benefits from all the projects would arrive in 20 days.

If we have only one of each resource, how should we proceed? Each resource will have to spend some time on each project. So, let's take our resources and multitask them. In Figure 9, the project managers for projects 1, 2, and 3 each conscientiously insist that some progress be made on their project each week. After finishing a task on one project, the active resource switches to another task on another project, so he or she (and the project manager) will have "progress" to report each week. The toll on the resources from such "project switching" is demanding — but we'll ignore it.

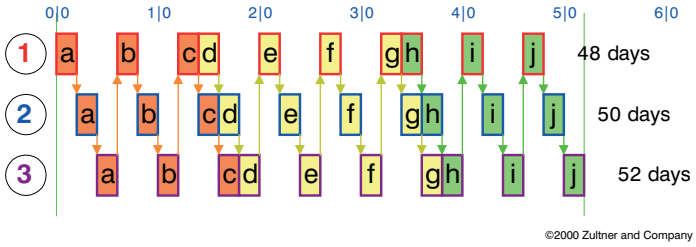

Figure 9 — Base case: three projects, multitasked resources.

The result is maximum resource utilization. All resources are working all the time. All projects are making progress. Of course, it takes longer when you don't have dedicated resources because one person cannot do the same amount of work as three people in the same amount of time. The projects finish in 48, 50, and 52 days. Customers will wait more than twice as long before they receive any benefits — at least 48 days. If they complain, what will we say? "Give us more resources! Everyone's working all the time already. There's no way we can deliver projects any faster than we are now with the people we have." (Is this a familiar refrain?)

Is there a better way? Let's see how critical chain manages a set of projects from a true systems perspective.

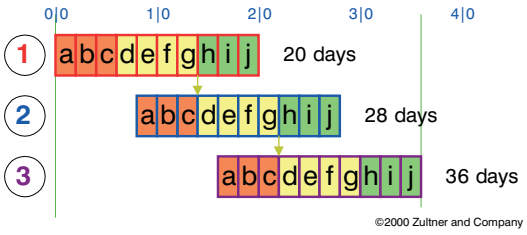

Figure 10 — Critical chain: three projects, no multitasking.

In Figure 10, each resource completes all tasks on one project before switching to another project, avoiding bad multitasking. Each project is finished in the same elapsed time as with dedicated resources. No resources switch

from one task to another and back and try to pick up where they left off. No resources are overloaded. This is how people prefer to work. This is how people do their best work, with the fewest errors.

The "efficiency" of multitasking is a myth.

The result is maximum project throughput. All projects finish sooner (20, 28, and 36 days). Even the third project, which waits 16 days to start, finishes 16 days earlier than the multitasking case. All projects are better off. The "efficiency" of multitasking is a myth. Just because a resource is utilized does not mean it is productive (or genuinely efficient). Here, we have lower resource utilization than with multitasking. Only the second resource, the designer, is working all the time. The first and third resources have a break of two days between each project — they are only 80% utilized! Yet we achieve higher project throughput and earlier delivery. Everyone wins.

The best way to manage a set of projects is for high throughput, not high resource utilization. And the best way to do that is to eliminate bad multitasking of resources. So, instead of getting three projects done in 52 days with multitasking, how much could be done in that time without multitasking?

In Figure 11, we see the impressive gains of critical chain multiproject management. With the same resources that were doing only three projects in 52 days (probably with some overtime), we can do five projects — at no additional cost (and with lower utilization, and therefore no overtime). This is real productivity, and we did this through changes solely in the project management domain. There were no changes to any aspect of software development methods or environment. We incurred no additional cost, hired no additional resources, and took no additional risks.

If you eliminated multitasking on all projects in your organization, you would reduce the elapsed time of all your projects by at least 15%-25%. For the small example above, with only three projects over only 52 days, the reduction in elapsed time of 16 days is 15%. You have more projects? Longer projects? Then the 15%-25% reduction cited here is *very* conservative.

Do Silver Bullets Exist for Software?

Some people say the example here isn't real. "Surely, it is not possible in practice to significantly reduce the elapsed time on all our projects without making any tradeoffs or any changes to how we build software. That would be magic — and we know that silver bullets don't exist for software." Well, the experience to date of more than 100 companies (software and otherwise) on five continents suggests that at least "silver BBs" (tiny silver bullets) do exist. Let's look at how such results are possible, without magic.

Start Later to Finish Earlier

By not doing bad multitasking, by staggering the projects, we accomplish more — even though we have lower resource utilization. We got two-thirds more project throughput by starting projects later. We complete the three planned projects in two-thirds the time by delaying the start of the second and third projects. Starting projects later means we finish earlier, because we are

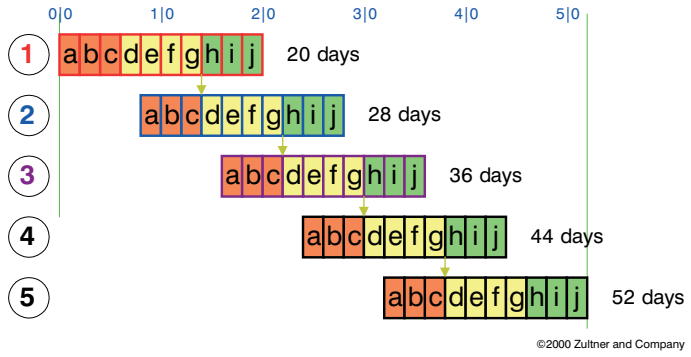

Figure 11 — The gain from critical chain.

managing a *set* of projects with shared resources. We have a *system* of projects.

Everyone can win, if we use a true systems approach to managing our set of projects. It is possible to eliminate bad multitasking *very* quickly. Would your customers, and your senior management, be interested in all projects taking less time and being delivered sooner without any additional costs or resources?

Critical chain accomplishes the "magic" of schedule reduction in ways that make sense but are decidedly nontraditional.

CONCLUSION

We have covered the basics of single-project critical chain and reviewed multiproject critical chain. We have seen that critical chain accomplishes the "magic" of schedule reduction in ways that make sense but are decidedly nontraditional. Critical chain is, in many ways, less complicated than many of the old project management practices it replaces.

Single-project CCPM can reduce the elapsed time of any individual project by at least 15%-25% through the efficiencies of pooled safety, which requires relay-runner behavior in project teams. Multiproject CCPM can also reduce the elapsed time of all projects in your organization by 15%-25% through the elimination of bad multitasking, which requires staggered project starts arranged by the management team.

The gains from each method are independent and therefore additive. You should do both, in one implementation effort, and reduce the elapsed time of your projects by amounts that will be difficult for people in other organizations to believe.

Today, some software organizations are already managing their projects this way. Someday, all software projects will be managed this way. When will your organization start using critical chain?

RESOURCES

The following are listed in order of recommended reading. The first three references are business novels that are enlightening and entertaining.

On Production
Goldratt, Eliyahu M., and Jeff Cox. *The Goal: A Process of Ongoing Improvement*, 2nd revised edition. North River Press, 1992.

On Distribution
Goldratt, Eliyahu M. *It's Not Luck*. North River Press, 1994.

On Project Management
Goldratt, Eliyahu M. *Critical Chain*. North River Press, 1997.

On Managing the Enterprise
Goldratt, Eliyahu M. *Goldratt Satellite Program* videotape series (eight three-hour tapes), 1999. Averham Y. Goldratt Institute Ltd., Amsterdam, the Netherlands. Available in North America from the International Institute for Learning in New York City.

On the Theory of Constraints
Goldratt, Eliyahu M. *Theory of Constraints*. North River Press, 1990.

Goldratt, Eliyahu M. *The Haystack Syndrome: Sifting Information Out of the Data Ocean.* North River Press, 1991.

Richard Zultner is a Senior Consultant with Cutter Consortium's Agile Project Management Practice and a contributor to that Advisory Service. He is a founder and director of the QFD Institute, a nonprofit research organization dedicated to the advancement of quality function deployment (QFD). His primary area of focus is efficient software process improvement approaches, including theory of constraints (TOC), daily management methods such as statistical process control, and cross-functional management techniques such as QFD.

Mr. Zultner received the Akao Prize in 1998 for his work in applying QFD to software. He is an international consultant, educator, author, and speaker and has professional certifications in quality, software quality, project management, software engineering, and TOC. Mr. Zultner can be reached at rzultner@cutter.com.

in the zone

A New Vision for Project Management

by Douglas R. Brandt

In the late 1990s, the Abbott Diagnostic Division (ADD) faced a very difficult business question. We asked ourselves, "How *effective* is our current project management approach, when measured against business results?" The all-too-painful answer was "not very effective."

The most visible sign that things weren't right? *Projects were not being delivered consistently on time*. We were making strategic and tactical business decisions based on project schedules, and the missed completion dates were neutralizing our business decisions. It eventually became a matter of importance to the success of the business unit.

It was time to take some action, so we began a serious search for a solution. We concluded that critical chain project management (CCPM) had potential for providing some of the solution to our problem. CCPM had enough research and real-world application behind it to be credible. There were practical tools that the business could use to improve project management results. In addition, it gave us the hope of a division-wide scalable system.

After interviewing several consulting organizations promoting CCPM, we selected a group to help us. In making our selection, we used the following criteria:

- We needed opportunities to test the solution and adapt it to our environment.
- We wanted a practical approach and help, not theoretical hand-waving.
- We wanted a complete solution that allowed us to achieve complete self-sufficiency.
- The group had to be professional, knowledgeable, and flexible.

As I write this, it is now over three years since our first pilot project using CCPM. The division has adopted the CCPM approach as its standard for all projects at all its locations.

CRITICAL CHAIN AND PROJECT MANAGEMENT

Many world-class organizations are proponents of CCPM, and it's quickly becoming recognized as a best-in-class project management methodology. CCPM is based on the theory of constraints (TOC). TOC is a fancy term behind a simple concept: when optimizing or modifying a system, you should focus on the few processes that create bottlenecks — the *constraints* of the system. Some of the most visible constraints in product development include limited resources, poorly allocated resources, and conflicts between tasks for shared resources.

There are additional facets of CCPM that make it a more practical approach than traditional project management. For example, there is ample research that systems are better protected from variability when the protection can be aggregated and strategically placed in that system. CCPM employs that protection in the form of buffers in the project schedules. Another key aspect of CCPM is explicitly integrating the concept of task engagement. By focusing on completing tasks in priority order, task engagement deals with the lost productivity that multitasking causes. When people are given clear and stable priorities and they understand the impact of switching tasks, more of the critical tasks will be worked from start to finish, speeding project completion.

In contrast to world-class organizations, many other companies continue to take project management for granted by treating it as a trivial matter. In these companies, there's a perception that anyone can put a

plan together and deliver a project or product on time, within budget, and with all the business expectations for the product met. Other companies take project management more seriously but may have seen project after project deliver late or not at all. Even in companies that take project management seriously, it is not uncommon — and possibly even the norm — to believe that project management is simply a matter of using project scheduling software. However, this assumption is untrue. In ADD, in addition to scheduling, project managers are responsible for:

- Delivering their projects on time
- Following government guidelines and regulations
- Implementing the division's policies and procedures
- Coordinating the diverse groups launching a product
- Overseeing design control and documentation processes

Excellence in project management requires business culture and behavior shifts that result in changing the way *everyone* in an organization acts.

In 2001, we formed the ADD Project Management Office (PMO) to focus our energy on more efficient product development by promoting and sustaining the behavior changes needed to implement CCPM. A team of professionals from three Abbott divisions was assembled to form the PMO at the Lake County (Illinois) site, and a consulting organization was retained to assist in this process. The purpose of this article is to discuss the mission, roles, and initiatives of the PMO and the results of this endeavor.

BEGIN WITH THE END IN MIND: THE VISION

Central to any journey is defining the destination. Though we may derive value from the journey itself, the goal of every journey is to reach a specific destination. Where do we think we're going with the implementation of critical chain? This question is addressed in Figure 1.

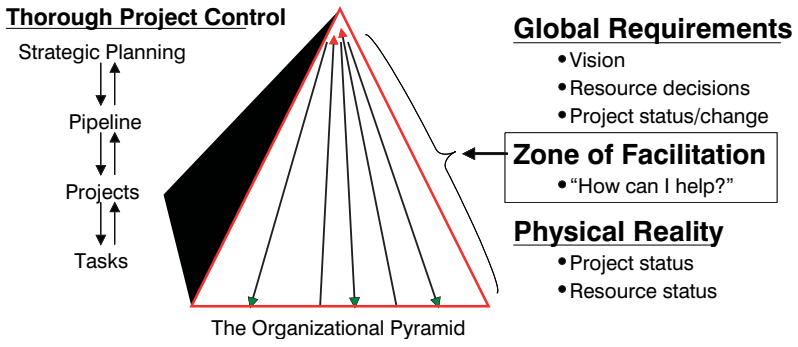

Figure 1 — ADD critical chain project management vision.

The vision of the PMO is to implement CCPM across all of ADD and to integrate this methodology from the individual task level up through the division's multiproject strategic plan. We've changed the name to Abbott Project Management (APM) to make it our own. The elements of achieving this vision are:

- Helping teams create credible project timelines that meet the business need
- Mentoring teams and management to adopt behavior changes that expedite task execution
- Using multiproject analysis to understand resource load issues and to "pace" projects based on the division's capacity

This process creates an environment in which global requirements are linked with physical reality and behavior changes result in consistent critical chain project timelines that are combined into a multiproduct pipeline model. In this way, the reality of activities "in the trenches" is captured in realistic individual project timelines, which feed the multiproject pipeline, and eventually this reality is reflected in the strategic plan.

Stated another way, this system ensures that the strategic plan is meaningfully translated into the work defined by each project timeline's individual tasks. We envision that the entire organization will become a "zone of facilitation," where managers and team members alike cooperate to speed

project work. The PMO seeks to be active across this zone by bridging the gap between high-level strategy and individual project activities. We want the business benefits that APM has the potential to deliver, we want to be self-sufficient, and we want to grow the application of APM by building upon each successful effort.

This kind of change is not easy, and when we implemented APM, we experienced a plethora of reactions throughout the organization. These reactions included everything from near hostility to outright joy. Initially, there were people who viewed APM as just another flavor of the week — they were skeptical and cynical. But now, even in the most formerly hostile environments, most members of the team recognize the value of APM once we have completed its implementation on their project. Once APM takes hold, it tends to stay in place in the organization, primarily because of the perception that it truly offers value.

More challenging for us were the behavioral and cultural obstacles we had to overcome. Like many big businesses, ADD had an unwritten set of project management norms. For example:

- Acting is better than planning.
- Project management happens within a function, not between functions.
- When someone wants something, it is always better to say yes than no.

In addition, we had many other requirements that applied to our work, and these requirements were often very complex. The result was that planning was difficult, cross-functional planning was even more difficult, and honest cross-functional planning was the hardest of all. These difficulties have led to some organizational resistance. However, we have found that most of the resistance is based on lack of understanding; when the concepts are understood, only a very small percentage of people maintain their objections. With our excellent implementation team, senior-level support, quality consulting help, and plenty of persistence, we are well on the way to making the revised norms — including planning, cross-functional cooperation, and the ability to say no when necessary — a part of our culture.

Planning was difficult, cross-functional planning was even more difficult, and honest cross-functional planning was the hardest of all.

THE PMO MISSION AND ROLES

The mission of the PMO is to ensure ADD's industry leadership by getting quality products to market first. We can achieve this by driving excellence in project management. We have identified the following roles for the PMO in implementing APM across ADD:

- Education, training, and technical support in APM
- Centralized project status communication (via the PMO intranet home page)
- Project management outsourcing from the PMO
- ADD project portfolio management (multiproject analysis)

PMO INITIATIVES

In the 18 months following the PMO's creation, more than 40 projects have implemented or are implementing APM. These implementations have involved multiple ADD sites in the US and all over the world.

Along with these implementations, the PMO has undertaken a number of major initiatives, many of which are accessible on the division's intranet site and which include:

- Creation of a database of active timelines, which are updated weekly
- Publication of APM success stories
- Weekly postings of multi-project "fever charts," at-a-glance diagrams showing project buffer status
- Construction of timeline templates, which improve consistency among projects' timelines
- Implementation of ProChain Enterprise, which allows managers to access timeline information with their Web browsers

The combination of the tools and the information resources above gives the project managers what they need to feel confident about their responsibilities going into every new product development effort.

CCPM enables teams to rapidly communicate the impact of potential task delays, develop contingencies, and still launch on time.

Throughout implementation, we have stressed that business culture change is required for improved timeliness in project completion. Important behavioral changes are discussed and emphasized in PMO training courses and include:

- Setting and communicating stable priorities
- Using timelines to see the gap between reality and ideal
- Practicing task engagement (working a task until done, minimizing interruptions)
- Running the "relay race" (ensuring rapid, clean handoffs of tasks)

Some of these principles are at first intimidating, especially the concept of task engagement in a company that prides itself on its employees' ability to "multitask." The training helps management and individual contributors to understand the basic principles of APM enough to be comfortable with them. Training reduces the inherent anxiety associated with change, especially in the high-pressure world of new product development.

APM RESULTS

"The proof of the pudding is in the eating. By a small sample we may judge of the whole piece."

— Miguel de Cervantes Saavedra, *Don Quixote*

And so it also goes for anything we do: the pudding may look good enough, but how does it taste? APM may sound good, but does it work? Here's some evidence for you to consider:

1. **Increased sales and profits.** The largest development project in the division, critical to our future, was reported by the project team to be three to six months ahead of where it would have been without APM. This translates into a multimillion-dollar upside.
2. **Increased on-time and early delivery.** The critical chain methodology and tools enable teams to rapidly communicate the impact of potential task delays, develop contingencies, and still launch on time. Before we used APM, projects were rarely completed early and many times delivered late. As of the end of 2002, of 33 completed projects scheduled with APM, 15 ended one to three months early, 12 ended within two weeks of scheduled launch date, and only six ended one to two months late.
3. **Increased predictability, visibility, and control.** With APM tools, it is apparent months before an end date whether a project is on track or not. Project status is determined on a weekly basis and displayed on a multiproject fever chart. APM enables us to take proactive measures to deal with the project issues. In contrast, traditional project management calls for remedial action only after a project is already in jeopardy.
4. **Increased quality.** Prior to implementing APM, it was not uncommon for a project manager to build a timeline by him- or herself, with a minimum of external input. This resulted in an inaccurate timeline of limited usefulness. With APM, it is standard for the entire cross-functional team to participate in timeline building and status updating, resulting in fewer missed issues and increased quality. The focus is to develop a clear understanding of what is needed before handoffs are made, not merely track task due dates. This improvement is not intrinsic to CCPM but is a result of good, old-fashioned, project management.
5. **Increased knowledge of processes.** The cross-functional communication described above leads to a dramatic increase in the whole team's understanding of the business processes and handoffs. This is good for employees and good for the division because this knowledge can be used

to improve our processes and shorten our cycle time. Again, this improvement is not due to the use of critical chain versus critical path but is due to a well-defined project management process.

6. **Improved morale.** On average, across 17 APM implementations that have been evaluated by the end of 2002, the teams' sense of overall project health increased from a score of 2.8 out of 5 before implementation, when they were using more traditional project management methods, to 3.9 out of 5 after implementing APM.
7. **Increased employee retention.** After implementing APM, voluntary separations in groups utilizing this methodology decreased approximately 50% compared to the division average. Surveys indicated that this improvement in employee retention coincided with a significant increase in employee quality of life due to the use of APM.

CONCLUSION

As detailed above, we have seen a significant improvement in our project management capabilities and project outcomes due to adopting a project management system, APM, that is based on CCPM. We believe that many of these improvements are due to employing the principles of the theory of constraints in our APM methodology. For example, items 1, 2, 3, 6, and 7 above are due to the behavior and business culture changes arising from implementing a system based on CCPM. In contrast, items 4 and 5 above are benefits of any rigorous project management method. We are currently linking our reality-based management system with our portfolio process and thereby aligning system capability to portfolio objectives.

We've had plenty of time to evaluate our process and results in implementing APM in our organization. We're self-sufficient, and we're well on our way to institutionalizing APM throughout the division. We're growing it organically with our own certified professionals. For us, some key success factors have been:

- Ensuring that the head of the business or business unit is a champion for the change that would be required to implement APM. Our senior VP in charge of R&D was a champion, and the other senior leaders were supportive.
- Making a group in each business unit responsible for implementing the change, reporting directly to the champion. Our PMO reports directly to the VP mentioned above.
- Thinking through the implementation plan very completely and carefully. We spent a lot of time with senior people on the R&D implementation plan.
- Making the implementation a key goal for all management personnel in the business unit. We did not do this initially, and the going has consequently been much slower than it could have been.
- Training the entire business unit in the business process changes and the business behavior changes that are required to succeed in the implementation. This is *not* something that should be done as quickly as possible; we've found that training people before they are ready to use the skills is worse than useless. They forget, but they don't believe they have forgotten.

Bottom line? APM is a viable tool for increasing revenue, decreasing costs, and improving quality in a business where the mission is new product development. The key criteria for success are: pick the right partner, make sure the organization is committed from top to bottom, and make institutionalization and self-sufficiency your objectives. Good luck.

Douglas R. Brandt is Director of the Abbott Diagnostic Division (ADD) Project Management Office. With more than 20 years' experience in the development of diagnostic assays and in project management, Dr. Brandt has championed the use of critical chain project management in ADD. Beginning in 1999, Dr. Brandt has worked to create a project management standard for the division that is based on the theory of constraints. His efforts have resulted in the application of this approach, Abbott Project Management, to the entire organization, including the division's four business units and 12 sites both nationally and internationally.

Dr. Brandt can be reached at E-mail: Doug.Brandt@abbott.com.

Segway and an Agile Critical Chain

by Matt Gelbwaks

THE SEGWAY HT

The Segway Human Transporter (HT) is the first of its kind — a self-balancing, personal transportation device that's designed to operate in any pedestrian environment (see Figure 1). It gives people everywhere the ability to move faster and carry more, allowing them to commute, shop, and run errands more efficiently. And it does it all by harnessing some of the most advanced, thoroughly tested technology ever created.

Figure 1 — The Segway HT.

To bring the Segway HT to life, a new company was established, and key outside suppliers, which would play an integral role throughout the development process, were brought on board. The Segway HT kept evolving, driven by creativity and the desire to build something that could make a difference. Developed mostly in secret, the product was revealed to the world in December 2001 and became available to commercial customers shortly thereafter.

Looking at early HT prototypes in the Segway Museum, one is struck by the clear evolutionary growth of the machine. The approach that Product Development chose to manage the execution phase was a hybrid in which an agile development methodology was used to manage the content of an iteration, and a critical chain scheduling (CCS) approach was used to manage how the iterations rolled up into the overall implementation. Though most documented agile efforts have been constrained to software development, Segway governed the entire product development effort — hardware, software, and mechanical components — with this blended approach.

CRITICAL CHAIN SCHEDULING

A critical chain schedule is a plan that is governed by the longest set of dependent activities (the critical chain). These activities are specified at their lowest level of complexity — the task. Each task has a well-defined scope whose effort can be estimated. All tasks are defined with explicit consideration of resource availability. They are scheduled with reference to the last date by which they can be completed without affecting the critical chain, rather than the earliest date by which the resource is available.

The advantages of executing a program using a CCS approach are primarily found in the way tasks are estimated and then scheduled into the plan. The concept of "Aggressive-but-Possible" (AbP) scheduling is used to size tasks by asking the staff how long it would take to complete the task if:

1. There were no other interruptions (single tasking versus multitasking)
2. The AbP estimate would define the schedule (no reestimating by management)

In most estimation exercises, staff members perceive that there are

distinct disadvantages or penalties for misestimating their tasks. It has become common practice to expect people to spend 100% of their time on a particular task but still attend meetings and events, most of which are not included in the project schedule. When estimates do not come out as management hopes, it is also a common practice to reestimate the tasks for the staff so that when the schedule rolls up, it is no longer than the desired duration. As a result, the original estimates become "padded," so that the tasks are still executable after management manipulation.

Once staff are confident that the two AbP rules stated above will be observed, subsequent time estimates shrink dramatically. Buffers are placed both within the project where the critical chain has dependencies on other tasks or resources and at the end of the project to support uncertainties within the critical chain itself. After all, there will always be things that come up that are not planned for, whether they be sick days or unanticipated complications in the execution of the task. Unless we can accommodate the unexpected, the entire plan will fail. Buffers are created to manage the uncertainties in both resource availability and technical execution.

Critical chain scheduling used in concert with a critical chain project management (CCPM) methodology has been able to demonstrate schedule gains of 25%-50% against traditional PM approaches. Despite these demonstrated gains, the IT community has not yet broadly embraced CCPM.

AGILE DEVELOPMENT

Agile development methodologies, such as Extreme Programming and the Crystal methodologies, have seen explosive growth within the IT community over the past five years. There has been much discussion of what types of development are appropriate for agile development, but when it is used, the resulting efforts seem to be characterized by high-quality software executed significantly faster than a standard development methodology would allow.

In an agile effort, the project is seen as belonging to the customer. Most customers know precisely what they want but have difficulty expressing it to the team responsible for execution. In standard methodologies, the development of requirements becomes an iterative and sometimes painful exercise. In an agile effort, this is overcome by only requiring the customer to have a well-defined objective. The project team takes a divide-and-conquer approach interactively with the customer so that the objective can be broken down into a set of prioritizable goals. These goals are defined as *stories*, each told from the point of view of the customer. The team defines the levels of effort necessary to achieve each story by dividing it into tasks, and then the customer has the opportunity to reprioritize based upon the defined schedule juxtaposed to current business needs.

Unless we can accommodate the unexpected, the entire plan will fail.

Although it would appear that scheduling would be very chaotic, the practice is to freeze each set of priorities until the short-term goals are met. The system grows organically though combination of the contents of each subsequent cycle. After each cycle, the integrated system is put aside (typically into use), and all the remaining goals are reprioritized. From this set, the customer chooses a finite number for development in the next cycle.

The schedule gains in an agile effort are achieved in several ways:

1. Estimates are conducted in ideal time — the amount of time it would take to complete a task if the staffer had no other things to do (meetings, vacations, etc.).
2. Estimates are accepted as offered. Peer pressure and personal accountability limit the estimates to no more or less than they need to be.
3. Cycles are of a constant size predefined at the onset of the cycle. Given the resources available, the customer knows for each cycle exactly how much will get done. If

content is more important than schedule, more cycles can be executed to meet the content goals. Otherwise, the customer can let go those tasks that are not completely necessary and maintain the schedule.

Clearly, both critical chain and agile scheduling approaches have much in common. It also should be clear that there is much that separates them. Neither approach is well utilized at this point, but both show great promise in speeding up development efforts. Agile development seems naturally suited to those projects where the system user cannot yet clearly express how the system will be used, and therefore which tasks may not even be necessary. CCS is most useful when all the tasks are clearly needed to achieve the ultimate project objective. Through the use of buffers, inherent technical risks can be incorporated into the schedule and management can be alerted when technical complications arise.

As we will see, our experience at Segway illustrates the benefits that can be realized when the two approaches are blended together.

We would have encountered difficulties using a pure CCS approach in the development of the machine because no specific endpoint was identified.

SEGWAY'S APPROACH

By the time we completed the HT's ideation phase, there were several working prototypes and a number of suppliers that were independently designing components for the machines. As it became time to execute the design, we wanted to move the implementation as quickly as possible. However, since the machine had never been used by the ultimate customers, it was somewhat difficult to predict what would be needed for public acceptance. There were several themes that were stated as company directives:

- Safety would be uncompromised.
- Quality would be absolute.
- Cost and performance would be optimized.

Executing these directives, we had to manage strict resource limitations due to the secrecy of the project, and we had the potential for technical complications since the system was well beyond the state of the art. Nonetheless, the program was undergirded by the strong objective of attaining inventor Dean Kamen's vision of a revolutionary device that would spark dramatic change in urban transportation.

The Hybrid

We would have encountered difficulties using a pure CCS approach in the development of the machine because no specific endpoint was identified. Rather than allow a schedule to dictate the release date, we determined that the machine would not be marketed until it was ready. This detail is extremely important in utilizing CCS, as the project network (rather than schedule) is created backwards from the endpoint and allows the schedulers to ask the question: "In order to get to this, what needs to be done first?" Thus it builds the schedule from the top down. Creating the project network with a sufficient degree of detail is a large effort for even small well-defined jobs, but once the network exists, it is very easy to identify the critical chain and the resource dependencies.

Once all the stories are identified in an agile development effort, only those necessary for the first iteration are detailed into tasks. This is important for two reasons. First, the implementation approaches and detail may be different depending upon the order stories are executed. Second, creating detailed specifications for every story is both time and resource intensive, and it may well be unnecessary should direction change during development.

For the Segway HT, we created a network of stories rather than tasks. The value here is that it was possible to start with the ultimate story — *Mass produce a Segway HT on our assembly line* — and then back all the way up to the initial stories such as *Complete proof of concept* and *Secure funding*. Once

this network existed, it was clear what the critical chain would be: advancing the state of the art sufficiently to make the product viable. All other chains of stories would be subjugated to this one … so the hunt was on.

The Segway HT was going to be manufactured inhouse, so the Assembly story could be created and controlled. However, we did not need to map out details until we knew which components would be included in the build. By using this hybrid approach and initially creating a program network, Product Development could identify the critical chain and determine the latest possible starting point for the Assembly story. When the initial proof-of-concept story was completed, Product Development was then able to identify several components that partners could develop and thus assist Segway in the HT's development.

Partnerships

As the partner arrangements and negotiations continued in parallel with the completion of the overall development story, Product Development was able to shorten the network by broadening it — doing more of the stories in parallel — and adding appropriate buffers to support the unknowns. It was not necessary to specify all of the implementation details to the partners. The partners accepted the stories and worked interactively to flesh out and approve their implementation approaches.

Michelin was Segway's chosen partner for the wheels and tires because of Michelin's knowledge base and broad experience. Product Development was able to focus on requirements, stating what operational characteristics were needed and then interactively engaging with Michelin's engineers as the tires were developed. The Segway engineer acted as the customer and managed the schedule buffer so that we would know how any perturbations might affect the overall program.

Personal Accountability and Buffer Management

The most critical aspect of getting the Segway HT into production on time was the personal accountability of everyone involved with the effort. Schedules were defined by consensus. The stories were discussed openly among the teams involved, and then individuals created and sized the task breakdowns. It was only once the sizing was completed that a schedule was created for the story.

Accountability is much easier to get from the team if what it is being held accountable for is something that it voluntarily created. Nonetheless, in true CCS form, a resource buffer was always added to the network to protect the team from unexpected situations. By managing with these buffers, it was possible to adopt openly people-friendly management approaches. Regardless of whether someone needed more time to complete a task due to unforeseen technical issues or family complications at home, the day-to-day impact to the schedule could be managed through the utilization of the resource buffer.

The most critical aspect of getting the Segway HT into production on time was the personal accountability of everyone involved with the effort.

The Product Development team took a low-tech approach to buffer management. Twice a week, there were standup status meetings at which the team discussed progress against the project network. All incursions into the buffers were identified, and as more of the buffer was utilized, the manager could review the affected tasks and easily comprehend whether this was a statistical fluctuation or a developing problem with the schedule. In this way, the managers were not always reacting (or overreacting) to progress issues due to the lack of visibility into their real downstream effects.

Theory of Constraints

This hybridized CCS approach has been one of the factors credited for our being able to launch the Segway HT into production so rapidly and with such a high level of quality. Though the introduction of the agile approaches may initially seem to be incongruous with critical chain methodologies, it is really much less so when viewed from an

overarching theory of constraints (TOC) vantage point. In TOC, only the bottleneck needs to be addressed to increase throughput, but by creating a network of stories and creating detail only for those tasks needed by the critical chain at the current point in time, attention can always be focused most directly upon the current constraint.

Barely 18 months passed from the time the design was completed to when the first units began rolling off the factory line.

With this hybrid agile/CCS approach, progress against the project network is clearly visible; the constraint moves up the chain, and with it, focus. When there are equally important dependent tasks and resources, the aperture can be widened to supply sufficient depth of field, allowing focus on each constraint. At the lowest level, the hybrid approach is based upon the five focusing steps from the theory of constraints:

1. Identify
2. Exploit
3. Subordinate
4. Elevate
5. Repeat

While the team maintains focus on the current constraint, other tasks can be completed in parallel without requiring undue attention until they become part of the constraint.

SEEING IS BELIEVING

It is hard to tell how much time the CCS approach has saved Segway, but considering that barely 18 months passed from the time the design was completed to when the first units began rolling off the factory line, the throughput of the process can certainly be appreciated. Beyond this, the degree to which we were able to achieve the corporate directives also adds perspective: safety is maintained with full redundancy in every hardware system and a field release defect rate of zero for all software systems. As Apple cofounder Steve Wozniak said when he was picking up his machine, "It is hard to believe this is the first generation of this machine." More amazing is that it is the first generation of the technology itself.

Matt Gelbwaks is responsible for product management and process implementations at Segway. Over the past 20 years, he has developed and consulted on methodology and development for the entire IT spectrum, from IT infrastructure to large real-time embedded systems. Mr. Gelbwaks is currently helping Segway refine its design and release infrastructure and roll out its product to the world. He has adopted the philosophy that no matter how esoteric or technically savvy your software (or hardware) is, if you're not able to deliver to the customers' expectations, there has been little or no value to the effort.

Mr. Gelbwaks can be reached at E-mail: matt.gelbwaks@segway.com.

get real

Bridging the Reality Gap

by Robert C. Newbold

LET THE GAMES BEGIN

Proponents of critical chain scheduling claim the power to revolutionize project management. They claim phenomenal results across many industries, and some say it is the most significant new idea in project management in the last 40 years.[1] And yet, in the "body of knowledge" documents of international professional organizations, project scheduling is acknowledged to be only a small piece of what project management is about.[2] Just as a revolution in a small country may not even make the newspapers in a large one, critical chain scheduling may seem insignificant next to the range of problems project managers face. In other words, what's the big deal?

To answer this question, it's first important to acknowledge that in the project world, there is frequently a gap between two different realities: the reality of senior leaders, who determine project requirements such as needed project completion dates, and the reality of the project team, those who need to supervise and conduct the project work. Typically the project manager and team members are stuck trying to resolve an impossible conflict — the "reality gap" between what is possible and what is required.

There are two ways to resolve this reality gap; I call them the "Old Game" and the "New Game." I will start by discussing the Old Game, which is how the project scheduling game is played in most large organizations today. This game results directly in a lack of credible project planning, which in turn leads to miscommunication and poor decisionmaking. Middle managers are left in a "zone of reconciliation," having somehow to reconcile dreams and reality through heroic efforts and creative storytelling.

I will then discuss the New Game. Credible critical chain schedules are fundamental to good communication and are therefore a key part of this game. However, the New Game can't be played just through scheduling; it requires process, discipline, and cooperation. When played properly, it results in a "zone of facilitation" across the entire organization in which stakeholders at all levels and functions work together to achieve excellent project results.

THE OLD GAME

The standard process for creating and using schedules (i.e., the Old Game) typically starts with requirements and dates. This information comes from senior leadership and/or marketing. In environments where schedules are created, the project manager then works with individuals to create a schedule that meets the requirements and the date. If there is an apparent conflict, the project manager prepares a position and negotiates with senior leadership to get something he or she thinks the team needs.[3] This usually involves relieving one or more of the big three constraints: resources, time, and scope. This flow is shown in Figure 1.

[1] See, for example, J. Cabanis-Brewin, "So … So What?? Debate Over CCPM Gets a Verbal Shrug from TOC Guru Goldratt," *PM Network*, Vol. 13, No. 12, December 1999, pp. 49-52.

[2] You can download various body-of-knowledge documents from the Association for Project Management (www.apm.org.uk), the Project Management Institute (www.pmi.org), and the International Project Management Association (www.ipma.ch).

[3] In further discussions, I refer to "senior leadership" when, depending on the situation, I may be talking about some combination of senior leadership, marketing, and the customer.

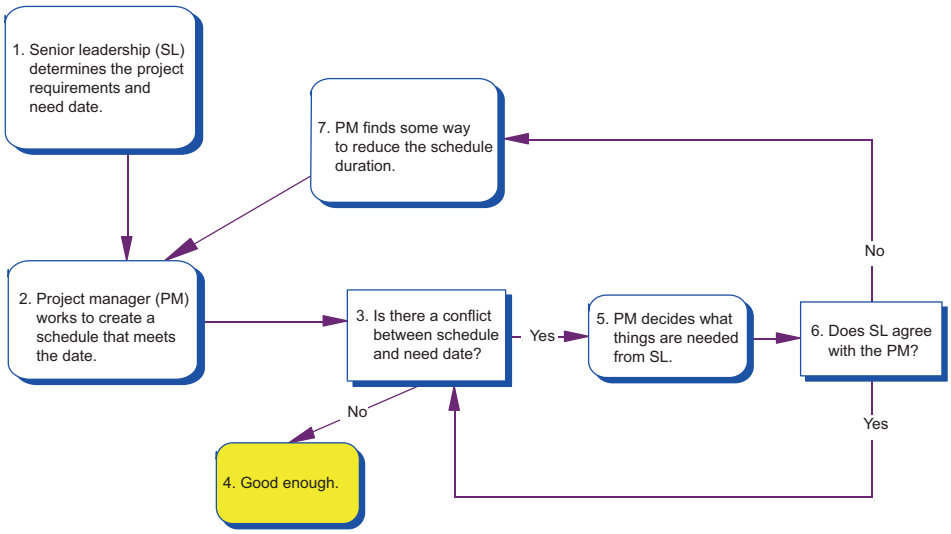

Figure 1 — "Old Game" playing board.

Rules of the Old Game

Organizations play the Old Game by putting new projects on box 1 and moving them to box 4 as quickly and painlessly as possible. The fundamental rule for everyone is individual survival, but there are others that vary from company to company. For example:

- If the requirements in box 1 are not impossible, management must be going too easy on the project team.
- In performing box 2, the project manager should avoid clarification of requirements. If you don't want to know the answer, don't ask the question.
- The meaning of box 3 — "Is there a conflict between schedule and need date?" — depends on what the meaning of "is" is.
- In box 5, ask for as much as you can, because you will get less than you ask for.
- Schedule credibility is irrelevant. The project manager should find the easiest possible way through box 7.

When playing the Old Game, it's common to put as many projects in play as possible. Often those projects assume availability of the same resources at the same time. There may be many projects that management doesn't even know about. As a result, most organizations seem to have far more projects in process than they can effectively work.

Note that boxes 1 and 6 are key control points for senior leadership. These are the points where leaders attempt to hold the line on resource use and project outcome. I have heard one senior leader, when presented with an alternative to the Old Game, ask plaintively, "If I'm sure the schedule is possible, how can I be sure people have an incentive to work as hard as they can?"

Winning and Losing

People can grow to be very skilled at the Old Game. They achieve organizational stature from their ability to win. Senior leaders can win by showing no mercy, pushing people to do more and more. Project managers can win by keeping something in reserve, always having tricks up their sleeve — or by moving on before the major problems surface.

Unfortunately, the three major project constraints are really four: time, cost, scope, and credibility. If the apparent conflicts between the team's perception and the business needs can't be adequately resolved, the project manager is put in a position where "unwarranted optimism" is the only answer. In some organizations, this is the first order of business: the schedule will meet the mandated dates whether or not the project team believes that makes any sense. This process produces a schedule that is missing several key components, such as:

- **Credibility.** There is a tremendous incentive to sacrifice honesty for expedience. As a result, few if any believe what the schedule is telling them.
- **Communication.** Without a solid, credible foundation for communicating project status and needs, no one has reasonable data with which to make decisions. Decisions must therefore be made based on intuition and hope.
- **Trust.** The Old Game tends to create an adversarial relationship between senior leadership and the project personnel. Senior leadership wants an earlier date and tends not to believe the project team's claims. The project team comes to believe that senior leadership isn't listening.
- **Ownership.** With lack of trust comes lack of ownership. The different players no longer feel that it is their responsibility to create success. The project manager tends to focus on arguments rather than results. The project team as a whole looks for excuses. Senior leadership becomes frustrated and less and less willing to help.

Over the long term, this lack of trust and ownership means that project managers and team members are frequently not contributing as effectively as they could. It also means that senior leadership is not supporting their efforts well. Often the project team's only viable decision is to work harder, while never finding the time to work smarter. The Old Game results in frustration, mistrust, and poor decisions. People struggle against one another rather than against the real problems. In the end, project performance suffers and everyone loses.

It's no wonder that predictability and visibility are so important to senior management. In my experience, most executives will put project predictability before speed on their organizational wish lists. This is a symptom of their lack of confidence and control. And yet the game that they play with their people virtually forces project communication that lacks visibility and predictive value.

THE NEW GAME

The "New Game" is played on the board shown in Figure 2. First I will

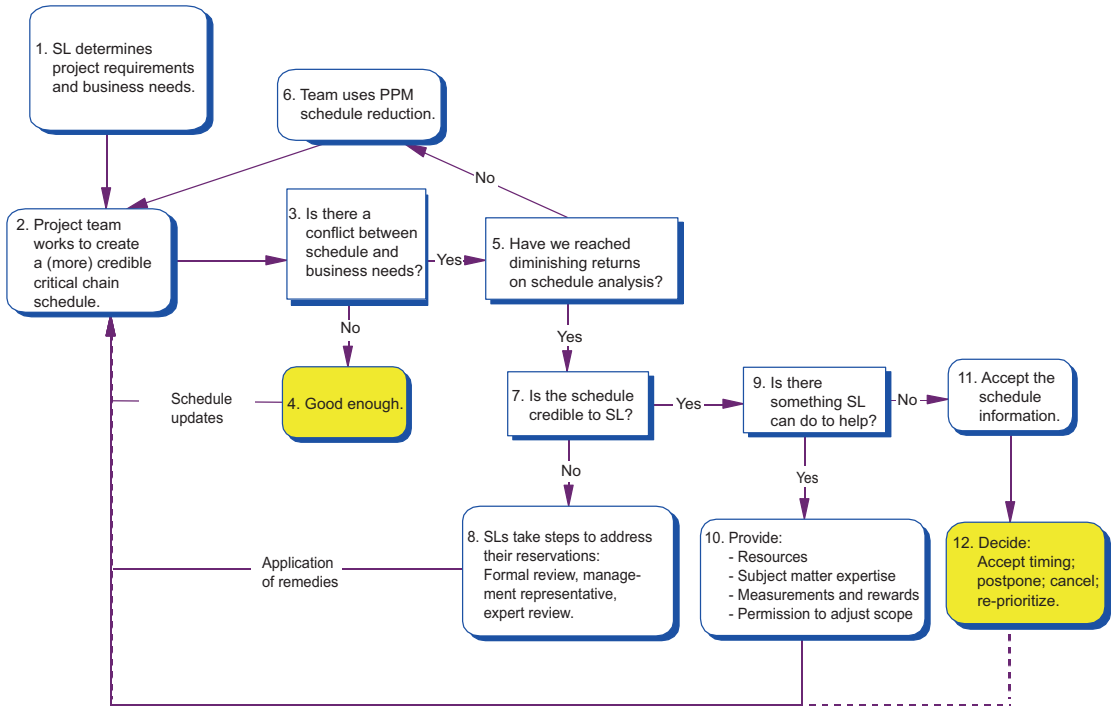

Figure 2 — "New Game" playing board.

describe the game board, then the new rules.

1. Senior leadership determines project requirements and business needs. Project requirements describe what the project needs to accomplish (what is done). Business needs include a description of why the requirements must be accomplished (what is the value to the company), when the project must be finished (the business need date), and the importance of the need date. This information should be well understood by the project team and written into a project charter that is approved by senior leadership. Otherwise, project teams may work hard to meet requirements that are not significant and ignore requirements that are crucial. Business needs may include budgetary considerations. Depending on the organization's gating processes, there may also be a feasibility phase during which a rough initial project schedule is created.

2. The project team works to create a (more) credible critical chain schedule. The project team must work together cross-functionally in order to create a unified picture of how the project charter will be carried out. The key is to have a credible departure point for analysis and decisionmaking. The initial schedule may or may not meet the date dictated by business needs. Credibility can be defined as playing by the following New Game rules:

- The project team is trying to present an honest picture.
- Risks are taken into account.
- Business objectives are taken into account.

We can track overall project progress much more effectively through consumption of buffers, and *everyone believes the results*.

My colleagues and I believe that critical chain scheduling is the simplest and most reliable method for creating credible schedules, both for single and multiple projects. Taking resource contention into account is an important part of the critical chain approach and certainly promotes realism. But the critical chain approach to protecting the schedule is essential to credibility. The traditional safety time put into task duration estimates to protect the completion of individual tasks is often "incredible," kind of like mandatory salary increases in contracts. Taking it out is unacceptable to workers; leaving it in doesn't work for management.

With the critical chain "buffer" concept, the safety time is taken out of the individual tasks and put back into the project plan in strategic locations as buffers. By managing this safety explicitly in buffers we still have protection, but it belongs to everyone. Everyone can see it and evaluate its use. We can track overall project progress much more effectively through consumption of buffers, and *everyone believes the results*. We have stable project schedules and predictable project outcomes. You could say that buffers move the focus of credibility from individual tasks ("I really need to get this task done on time") to the project ("We really need to get this project done on time"). Some tasks will be early and some late, and this is expected; the project itself, however, is very likely to be completed by the commitment date. Therefore the project schedule is truly credible.

As the New Game is played with more and more projects, it also becomes more and more important to understand interproject relationships. Credibility may also require taking into account the impact of multiple projects on shared resources, shared integration points, and relative project priorities.

3. Is there a conflict between schedule and business needs? The team must analyze whether the schedule is in conflict with the business needs. As part of the analysis, the team needs to resolve any lack of clarity about those needs. This question may require a review by senior leadership, especially if the schedule is ready for acceptance (i.e., there's no conflict).

4. The team reaches "good enough." Declaring "good enough" is always "for now." Initially the project team should brief senior leadership so that

leaders understand and approve the decision to move ahead with the schedule. The line from box 4 back to box 2 indicates that the schedule will always be under examination for improvements or needed changes. The "Schedule updates" note on the arrow out of box 4 implies that the schedule must stay credible; that is, it must be kept up to date.

5. Has the team reached diminishing returns on schedule analysis? There is sometimes a tendency for a project team to spin its wheels, unable to resolve the conflict between business needs and its perception of reality. There are many ways to help teams challenge their assumptions, such as:

- Ask, "If you could do whatever you wanted, with no resource or policy restrictions, what would you do differently?" This may result in the need for help from senior leadership, meaning you must first pass through boxes 7 to 10.
- Check the practices of other companies or groups inside the company.
- Provide an outside expert to review assumptions.

6. The team uses critical chain tools to reduce the schedule duration. A focused analysis approach using critical chain software allows easy identification of those tasks, links, and resources that are causing the project schedule to be as long as it is. Team members challenge one another's assumptions to reduce project schedules. During this process, it is important to have a cross-functional team that is familiar with both the critical chain concepts and the project itself. As usual, credibility must be maintained throughout.

7. Does senior leadership think the schedule is credible? Assuming the team has been careful to maintain a credible schedule, and the schedule doesn't yet meet business needs, we must be sure that senior leadership buys into the current picture. That doesn't mean it's the final schedule, and it doesn't mean senior leaders acknowledge that it is telling any final truth. It does mean the assumptions behind the schedule are understood well enough that senior leadership believes that — barring significant further interventions or understanding — the schedule is realistic. It means that senior management is willing to work with members of the project team (instead of, too frequently, against them), using the schedule as a point of departure.

8. Senior leaders take steps to address their reservations. If senior leaders don't believe the project team, whose problem is it? The common answer, the Old Game answer, is that it is the project team's problem. That answer effectively blocks further communication because *the project team is already stuck*. Realistically, the team is likely to teleport over to box 7 of the Old Game. The team members need help, not more pressure.

There are a number of steps senior leaders can take to cement their own belief, such as:

- Hold a formal schedule review to discuss schedule assumptions
- Put a trusted management or customer representative on the project team
- Have an outside expert evaluate the current project plan and make suggestions

9. Is there something senior leadership can do to help? Think of things that senior leadership might be able to do or supply that will help get a project done more quickly. There are many possibilities to consider, for example:

- More or better people
- Bringing in outside subject matter experts
- Changes to measurements and rewards (e.g., team incentives, spot bonuses, overtime pay)
- Permission to adjust scope
- Help finding ways that enable people to be more fully engaged in their work (e.g., quiet places to work)
- Making sure the project has sufficiently high priority

Some of these ideas, such as measurements and better work engagement, might profitably become a more global part of the workplace.

10. Senior leadership actually provides help. It's one thing to promise resources or other help when the project is being scheduled; it can be quite another to provide that help when the time comes. If the help is not provided, the assumptions behind the

schedule are invalidated, and the team's commitment (the schedule) becomes infeasible. This is true of behavior changes as well as resources. For example, suppose team members are encouraged to stick to their priorities and say "no" to lower-priority work. If senior leaders then criticize them for that, the desired behavior change (staying on task) is undermined. Any schedule assumptions dependent on that behavior, such as "no multitasking," may become invalid.

11. Everyone accepts the schedule information. There comes a point where further discussion appears fruitless. Everyone concerned must accept that the schedule is as good as it's going to get, at least for now. Senior leadership must make a decision. Sometimes this isn't easy; deal with it.

12. Senior leadership makes a decision about the project. Senior leadership, with the support of the project team and information from the schedule, must choose from among several possible alternatives. For example:

- Accept the timing — allow the project commitment date to be later than the declared business need. Even here, all is not lost; the team may continue to find ways to improve the situation.
- Put the project on hold while working on better opportunities and/or looking for partners, technology improvements, and so on.
- Outsource significant parts of the project.
- Rescope the project.
- Cancel the project.

Rules of the New Game

The object of the New Game is to get bottom-line results for the organization while maintaining credibility, communication, trust, and ownership. We may get to box 4 or box 12, but even with a "good enough" schedule, the journey is far from over.

There are several rules to the New Game:

1. Insist on honesty. It is the only basis for long-term trust. All players should give realistic need dates, requirements, task times, and so on.

2. Need can't win over reality. Denial takes you back to the Old Game. Senior leadership may not say, "This schedule is unacceptable" without being willing to help come up with alternatives. Project managers and team members should be rewarded, not punished, for reporting what they believe to be the truth.

3. Account for risks. No significant increase of risk (or decrease of buffer) is allowed to meet dates without being explicit about the magnitude of the risk, mitigation plans, and contingency plans.

4. All players should be asking, "How can I help?" Everyone should be focused on helping achieve the business results.

5. Focus on business objectives. Don't add task due dates to put more pressure on people; don't bear grudges or ill will.

6. Use the process on the game board. For example, doing anything without clarifying the business needs (box 1) is a mistake. How can the team make effective tradeoffs without a good understanding of the business needs? Jumping to senior leadership help (boxes 9-10) before establishing credibility (boxes 7-8) is a mistake. Senior leaders should never invest in a project when they don't believe the data justifies the investment.

7. Hold people accountable for following the rules.

"But wait," you may be asking. "If this revolutionary critical chain approach produces shorter schedules, why do we even need boxes 7-12? In fact, why do we need to play the New Game at all?"

Applied properly, the critical chain approach produces shorter *credible* schedules. Projects are completed much more quickly *relative to what would have happened otherwise*. The schedules may or may not be shorter relative to the arbitrary dates and schedules that arise from the Old Game. If you apply critical chain scheduling and senior leaders complain that the schedules are too long, you must proceed to box 7. Otherwise you will be back to the Old Game.

Winning with the New Game

The New Game is not about heroism and individual valor; it is about people working together to achieve great results. Even "ordinary" project managers can play the New Game with extraordinary success. Winning must be expressed in terms of stakeholders achieving

what they need. By winning with the New Game, the organization builds a culture that values things important to its long-term success, such as:

- **Credibility.** Not only is the project schedule credible, but the team is more credible to itself and to senior leadership.
- **Communication.** The schedule gives a basis for all project stakeholders to communicate effectively about the project and its requirements.
- **Trust.** Through good communication, senior leadership and team members build trust and overcome years of poor communication and dysfunctional coping mechanisms.
- **Ownership.** When all the key players have a say in what the project is and how it will be executed, and when they trust that the commitments are possible, they develop phenomenal ownership and team spirit.

After trying the New Game, one functional manager said, "For the first time, I really believe what the schedule is telling me. This is the first time that I've ever felt like we really have a chance to make it." A senior leader who learned to play the New Game well said, "I can talk to teams using critical chain software, ask questions, and get real answers. When there is a delay, they know what is causing it. The conversations don't break down to guessing and defensiveness like they used to." These are typical comments, but they are not a result of scheduling alone. They are a result of effectively integrating scheduling into the organization's business processes — the New Game.

CONCLUSION

The Old Game comes about through both the lack of credible schedules and the lack of good processes for using them to analyze and communicate effectively. All the standard complaints about project results — late, over budget, under scope, mistrust, burned-out people, and on and on — are also standard outcomes of the Old Game.

The New Game bridges the gap between business desires and project reality. It produces results vastly superior to those of the Old Game, as shown in Table 1.

The results of the critical chain approach plus the New Game have the power to revolutionize project management in any project organization.

Robert C. Newbold is CEO of ProChain Solutions, Inc. Mr. Newbold has more than 25 years' experience as a consultant, analyst, manager, software developer, and entrepreneur. He has developed and implemented breakthrough solutions for numerous Fortune *500 clients. In 1996, he founded ProChain Solutions in order to develop improved tools and processes for managing projects. ProChain's primary product is a system called ProChain Project Management, an end-to-end integrated solution consisting of methodology, software, training materials, and business processes.*

Mr. Newbold has authored numerous articles on the subject of project management. His book Project Management in the Fast Lane: Applying the Theory of Constraints *was published in 1998 by St. Lucie Press. Mr. Newbold speaks frequently on the subject of project management and critical chain for corporations and industry groups. He holds degrees from Stanford, Yale, and SUNY Stony Brook.*

Mr. Newbold can be reached at ProChain Solutions, Inc., 12910 Harbor Drive, Lake Ridge, VA 22192, USA; Tel: +1 703 490 8821; Fax: +1 703 494 1414; E-mail: rnewbold@prochain.com.

Table 1 — Comparison of Old and New Games

Characteristic	Old Game	New Game
Value from schedules	Minimal	Predictability, visibility
Schedule credibility	Low; planning is often abandoned as a waste of time	High; credibility is a requirement of the New Game
Communication	Poor, due to dependence on intuition and poor data	Honest, based on a shared perception of reality
Trust	Low	High
Ownership	Low	High
Triple constraint (time, resources, scope)	Often violated	Seldom violated
Reality gap	Significant	Closed

Cutter IT Journal

The Journal of Information Technology Management

Get global perspectives and solutions to some of the most critical business-technology issues facing organizations today!

Your Front-Row Seat to IT Management Debate at the Highest Level!

Every day your organization is confronted with the stark reality of having to achieve more aggressive goals with a shrinking budget, ever-changing requirements, and impossible deadlines.

Few of you have the time to develop well-supported arguments on how to get your organization to improve its IT operations. It's a tough trap: you know solutions are out there, but you're too busy to identify them and convince your organization to implement them.

Advice, Solutions, and Experience That You Can Rely On

A ***Cutter IT Journal*** subscription helps you break out of the trap. Every month, ***Cutter IT Journal*** features a select guest editor who articulates the controversial issues, offers his or her opinion on them, invites others to introduce opposing viewpoints, and sparks a lively debate.

Cutter IT Journal provides you the opportunity to experience a variety of perspectives: viewpoints that will be instrumental in advancing the cause of better software development. No matter where you stand on these issues, the thoughtful discourse delivered in ***Cutter IT Journal*** will certainly help you clarify your position.

In addition to your monthly journal issue, you will also receive the weekly ***Cutter IT E-Mail Advisor***. Each ***Advisor*** brings you practical advice and thoughtful analysis from well-known and respected experts in the information technology field. Learn from the experiences of others, including what you should avoid and what you should consider implementing.

As a subscriber to ***Cutter IT Journal***, you will stay up to date on important IT issues such as software quality and project management, security, risk management, business intelligence, sourcing, enterprise architecture, requirements, trends in technology, and more. Whatever the topic, you can be sure you will receive frank, unbiased opinion, in the no-holds-barred manner ***Cutter IT Journal*** is known for!

Don't miss upcoming issues on:

- Testing the Limits of Agility
- Service-Oriented Architecture and Governance
- Making Strategy Work
- Collaboration
- Business Intelligence Trends
- Out-of-the-Box Agile Applications

Special Offer! Begin your subscription to ***Cutter IT Journal*** today and **SAVE $100** off the regular subscription rate! Just complete and return the form below by fax to +1 781 648 8707 or call +1 781 648 8700. Or order online in our secure bookstore at **www.cutter.com/bookstore.html** and enter Priority Code 180*D01INS to receive your discount.

For more information on ***Cutter IT Journal***, please visit www.cutter.com/itjournal.html.

Special Offer: Save $100 on a New Subscription!

CUTTER CONSORTIUM

☐ ***YES!*** Please start my new, one-year subscription to ***Cutter IT Journal*** for just $385 (US $485 outside North America) — I SAVE $100 off the regular price of $485/US $585!

Name	Title
Company	Dept.
Address/PO Box	Mailstop/Suite
City	State/Province
ZIP/Postal Code	Country
Phone	
Fax	
E-Mail	

Fax to +1 781 648 8707, call +1 781 648 8700, or send e-mail to service@cutter.com. Mail to Cutter Consortium, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA. Order online at www.cutter.com/bookstore.html (enter Priority Code 180*D06PRO to SAVE $100).

Cutter IT Journal

Topic Index

March 2003	Critical Chain Project Management: Coming to a Radar Screen Near You!
February 2003	XP and Culture Change: Part II
January 2003	Ending "Garbage In, Garbage Out": IT's Role in Improving Data Quality
December 2002	Preventing IT Burnout
November 2002	Globalization: Boon or Bane?
October 2002	Whither Wireless?
September 2002	XP and Culture Change
August 2002	Plotting a Testing Course in the IT Universe
July 2002	Confronting Complexity: Contemporary Software Testing
June 2002	B2B Collaboration: Where to Start?
May 2002	Information Security and Privacy in a Fragile World
April 2002	Web Services: "You Say You Got a Real Solution ..."
March 2002	The Technology Myth in Knowledge Management

Bill Lynch, Guest Editor

Bill Lynch is President and Chief Operating Officer of ProChain Solutions, Inc. He has spent more than 15 years focusing on breakthrough process improvements. Mr. Lynch cofounded ProChain Solutions in order to focus on creating a company that would provide an integrated, innovative solution for product development and project and resource management. ProChain's primary product is a system called ProChain Project Management, an end-to-end integrated solution consisting of methodology, software, training materials, and business processes.

Mr. Lynch has an undergraduate degree in electrical engineering and graduate degrees in liberal arts and logistics management. He can be reached at blynch@prochain.com.

CUTTER IT JOURNAL

Upcoming

Issue Themes

Project Portfolio Management

Open Source

Enterprise Architecture Governance

The New CIO Agenda

Usability

Patterns

Killing IT Projects