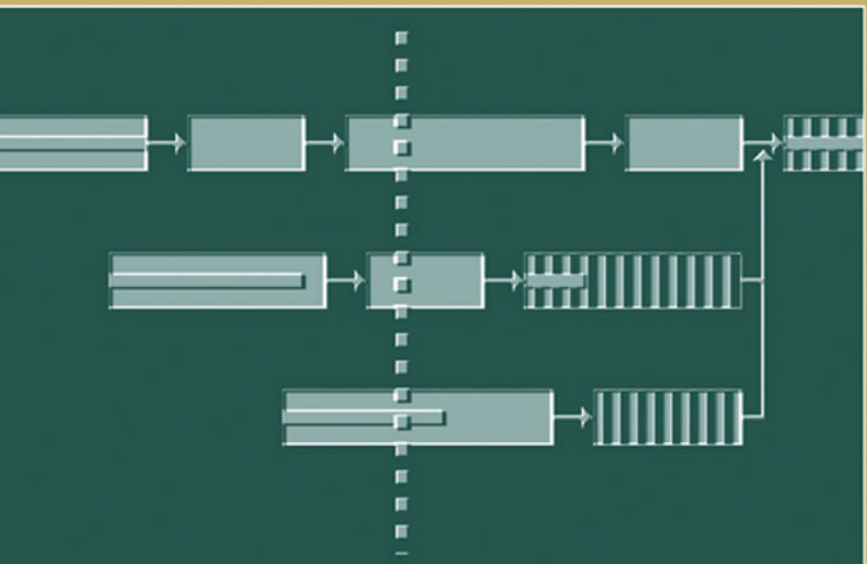


Lawrence P. Leach

Critical Chain Project Management

Second Edition

EFFECTIVE PROJECT MANAGEMENT SERIES

Critical Chain Project Management

Second Edition

For a listing of recent titles in the *Artech House Effective Project Management Library*, turn to the back of this book.

Critical Chain Project Management

Second Edition

Lawrence P. Leach

ARTECH HOUSE
BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data
Leach, Lawrence P.
Critical chain project management/Lawrence P. Leach.—2nd ed.
p. cm. — (Artech House effective project management library)
Includes bibliographical references and index.
ISBN 1-58053-903-3 (alk. paper)
1. Project management. I. Title. II. Series.

T56.8.L34 2004
658.5—dc22

2004046244

British Library Cataloguing in Publication Data
Leach, Lawrence P.
Critical chain project management. —2nd ed. —(Artech House project management library)
1. Project management
I. Title
658.4′04

ISBN 1-58053-903-3

Cover design by Gary Ragaglia

685 Canton Street
Norwood, MA 02062

International Standard Book Number: 1-58053-903-3
Library of Congress Catalog Card Number: 2004046244

10 9 8 7 6 5 4 3 2 1

Contents

Preface

When critical chain was first introduced, I was fortunate to be able to see instantly how it constructively added to the Project Management Body of Knowledge (PMBOK™), and worked to define Critical Chain Project Management (CCPM) as the marriage of these two knowledge areas. There was not much overlap between people familiar with the Theory of Constraints (TOC) and people in the formal project management community, in part because TOC started in the domain of production.

When the formal project management community first became aware of CCPM (in part due to my 1998 presentation to a PMI annual meeting in Long Beach, CA), they seemed to take one of two views. The majority was excited about a new approach, eager to learn it, and willing to apply it. A small but vocal minority put forth a resistance movement, generally voicing the view that "It's nothing new." Many letters crossed path in the PMI's publications PM Network and PM Journal, and several articles (including my own) helped define some key issues. I was eager for constructive critical thinking, but am unimpressed by arguments such as "it has been done before." Fortunately, cooler heads prevailed in the Project Management Institute (PMI). PMI agreed to sell my book in the bookstore, and sponsored my 2-day seminar, which played to full houses and received high praise from the attendees.

Now, several of the world's largest companies and major government organizations have proven the unprecedented project results achievable with CCPM. Letters to the editor of the PM Network have dwindled away, and the 2004 edition of the PMBOK™ Guide due out in a month promises to formally include critical chain as a scheduling method. Most of the new project management books contain critical chain, and I have been invited to contribute chapters to a few. PM Journal articles continue, but now present some real critical thinking that helps the process of continuous improvement. Although some old school project managers still confuse buffers with float, seemingly failing to appreciate the probabilistic thinking underlying CCPM, critical chain seems to have crossed the chasm into the mainline of project management.

Despite thousands of successful CCPM projects in a wide variety of project types and industries, CCPM has yet to become the standard in industry. It still qualifies as a new technology introduction. My consulting practice over the last decade has revealed a surprising lack of understanding basic project management in many companies. For some of those companies, CCPM has been the key to unlocking the entire world of professional project management. Others, already well versed in

conventional project management, have moved to reap the rewards of CCPM. But, CCPM is still in the early adoption stage of a new technology.

I invite you to consider CCPM as step towards improving your quality of life, and that of all of your project stakeholders. I invite you to partake of the benefits CCPM offers, including more predictable project success, shorter project duration, greatly improved organizational project throughput, and, most importantly, reduced stress and increased success for all. As you do so, I ask you to share your experiences with others so they can partake of the benefits, and help all develop even better ways to enhance project success.

Acknowledgments

Most of all I extend my appreciation to the early adopters of CCPM, who often went against conventional wisdom and their own organizational culture to create remarkable results. Although their organizations achieved great benefits, in many cases it required a major struggle to make to make the transition.

The credit and my appreciation for inventing the Theory of Constraints (TOC) and single-project critical chain goes to Dr. Eliyahu Goldratt. My gratitude also extends to Dee Jacob of the Avraham Goldratt Institute for introducing me to it in the mid-1990s. I am also thankful to Tony Rizzo for recognizing the multiple project approach, and for inviting me to work with him on some of the early implementations of CCPM.

CHAPTER 1

Begin in the Beginning

Projects fail at an alarming rate. Quantitative evaluations show that as many as 30% of projects are canceled before completion, wasting all the time, money, and effort spent on them. Surviving projects often fail to deliver the full initial project scope, or they deliver late and/or overrun the budget. Project delays and overruns frequently run to hundreds of percentage points. These failures consume billions of dollars per year. They occur in all cultures and for all kinds of projects. Attempts to improve project performance create personal and organizational pain and paperwork, with little positive, or even with negative, impact on project performance. The field of project management has not kept pace with improvements in other areas of human endeavor, such as technology and manufacturing. This book seeks to put you and your organization on a path to radically improve project success.

The first three chapters provide the background for Critical-Chain Project Management (CCPM), so if you are anxious to understand what it will do different for a single project, you can jump to Chapter 4. If you are even more anxious to start a single project, you can start with Chapter 6 on developing a single-project plan. Chapter 7 will guide you in planning multiple projects that share common resources.

This chapter provides the context for CCPM, starting by defining the problem and showing some data to assert that CCPM is proving to be an effective solution in a wide range of project types and industries. The main point of this chapter is to convince you that just working harder to execute conventional project management is not likely to give you the results you want, and to prepare you for Chapter 2, which develops a firm basis for change in solution direction offered by CCPM.

The Project Management Institute's (PMI) *Guide to the Project Management Body of Knowledge* [1] (PMBOK™ Guide) defines a project as "a temporary endeavor undertaken to create a unique product or service." The word "temporary" distinguishes projects from production-like endeavors. "Unique" means that projects are all different from one another. Project success means giving the project customers what they want, when they want it, for a price they have agreed to, and having a project team that is happy about creating that success.

Chapters 1 through 3 refer to the existing project system. Although change is underway, most of the existing project-management literature still primarily describes the Critical-Path Method (CPM) to define a project schedule. The PMBOK™ Guide alludes to other methods, and the 2004 edition promises to include critical chain, but CPM is the method used most, by a wide margin. Most

commercial software claims to implement CPM. The PMBOK™ Guide describes methods to deal with uncertainty in projects through consideration of project risk. It also describes the earned-value method of project measurement and control. Many large projects use project-risk management and earned value, especially on projects performed for the U.S. government. Although it is not a specific point of guidance, most software and all of the applications we have seen apply CPM using early-start schedules. This means that the software schedules activities as far to the left (or as early as) possible. Figure 1.1 illustrates a typical project plan using this method.

People sometimes also distinguish projects from production operations by the quantity of the products produced and the relative amount of time on task. Projects usually produce a one-of-a-kind result. Production operations produce many items, all more or less similar. There is a gray area between custom-made production operations (e.g., built-to-order automobiles) and projects. I have found that many people consider production operations and projects as distinctly different. In the mid-1990s, I first learned of a system theory called the Theory of Constraints (TOC), first described by its inventor, Dr. Eliyahu Goldratt, in his book *The Goal* [2]. I recommended this book to other program and project managers, only to find that they could not see any relevance of the book or the theory to projects. Subsequently, I discovered a method to break the paradigm. I draw Figure 1.2, and ask people, "Which is this, a project or a production operation?" The reaction is quite interesting. Most people look puzzled at first. They do not respond immediately. Then, someone finally offers up, "Well, it could be either." Others then promptly agree. Indeed, it could be either. At this level, the similarity is more striking than the differences. The primary similarity we are going to explore in this book is the connection of dependent process steps with variation in the time it takes to convert the input to the output of each step.

The actual time on task, or touch time, in production operations is usually a very small part of the delivery time. Many people claim that the actual time on task determines the overall time of the project and, therefore, approaches 100% of the project delivery time. Critical-chain questions this assertion.

1.1 Project Success

Successful projects meet the needs of everyone who has an interest in the project: the stakeholders. All projects have a goal. Figure 1.3 illustrates that satisfying the goal normally requires satisfying three necessary conditions. The scope sets a minimum standard for the project results. Necessary cost and schedule conditions usually set maximums. Figure 1.3 also illustrates resources in the center, with a relationship to all three necessary technical conditions. Project resources influence all three necessary conditions for success.

The three necessary conditions are interdependent. The longer a project takes, the more is costs. The more a project costs, the longer it takes. The longer a project takes, the more opportunities exist to change the scope. The more changes to the scope, the more costs and schedule increase. Subsequent definition of the project system explores these relationships in detail.

ID	Task Name	Duration	Start	Finish
2	Plan	5 days	Mon 6/2/97	Fri 6/6/97
3	Permit	5 days	Mon 6/9/97	Fri 6/13/97
4	Site Prep	10 days	Mon 6/16/97	Fri 6/27/97
5	Hole	10 days	Mon 6/30/97	Fri 7/11/97
6	Landscape	15 days	Mon 8/18/97	Fri 9/5/97
7	Drive & Walks	20 days	Mon 7/21/97	Fri 8/15/97
8				
10	Foundation	5 days	Mon 7/14/97	Fri 7/18/97
11	Frame	10 days	Mon 7/21/97	Fri 8/1/97
12	Roof	10 days	Mon 8/4/97	Fri 8/15/97
13	Sheath	10 days	Mon 8/4/97	Fri 8/15/97
14	Trim	10 days	Mon 8/18/97	Fri 8/29/97
15				
17	Plumbing	10 days	Mon 8/4/97	Fri 8/15/97
18	Electrical	10 days	Mon 8/4/97	Fri 8/15/97
19	Cabinents	5 days	Mon 8/18/97	Fri 8/22/97
20	Drywall	5 days	Mon 8/25/97	Fri 8/29/97
21	Paint	5 days	Mon 9/1/97	Fri 9/5/97
22	Trim	5 days	Mon 9/8/97	Fri 9/12/97
23	Complete	0 days	Fri 9/12/97	Fri 9/12/97

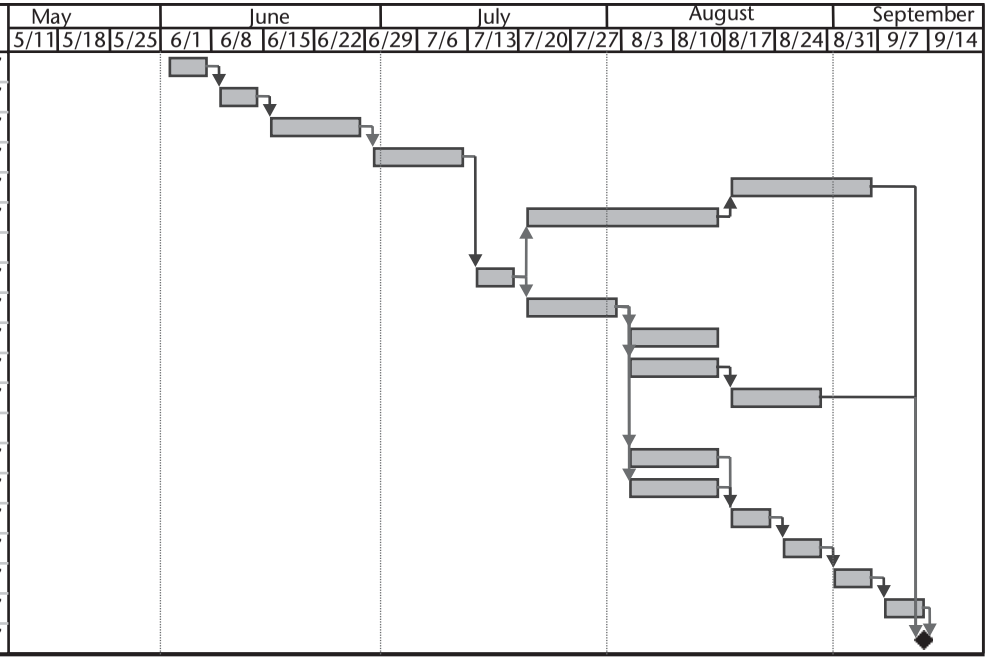


Figure 1.1 A typical CPM project plan identifies the critical path and activity early and late start and finish dates. Most of the time, project plans default to an early-start schedule.

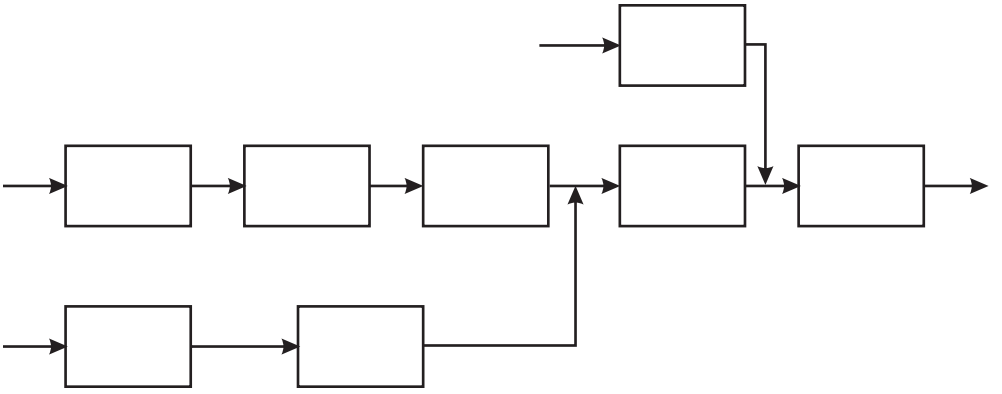


Figure 1.2 Is this a project or a production process?

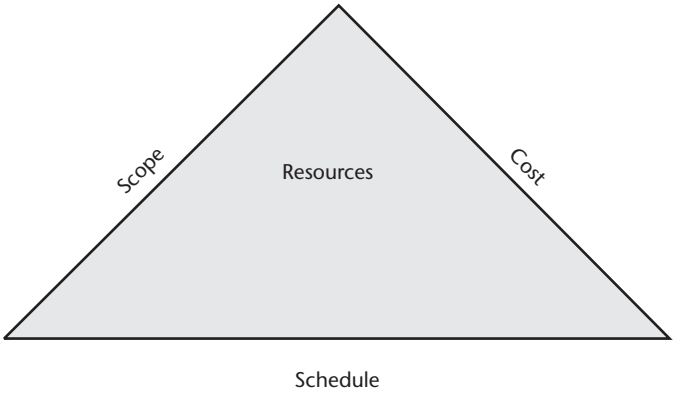


Figure 1.3 Satisfying the project goal requires three necessary conditions.

1.2 Defining the Problem

Most scientists agree that precise definition of a problem is the most important step to a successful solution. Karl Popper [3], my favorite philosopher, notes, "science begins with problems, and proceeds from there to competing theories which it evaluates critically." This text deals with the general problem of improving project success. Following Popper, I invite you to evaluate critically what I have termed the *present system*, or any system you presently use, as compared to CCPM. As you will see, the problem definition "improve project success" is a bit too broad to guide developing a systematic, effective solution.

1.2.1 How Good Is the Current Project System?

Ask yourself the following questions:

1. How often do you hear of projects taking longer than scheduled?
2. How often do you hear of projects completed much more quickly than originally scheduled, without a lot of expediting and pressure on the project team?
3. How often do you hear of projects going over budget?

4. How many projects do you know of that were completed for significantly less than the original proposed budget?
5. Have you ever heard of projects having to redefine their scope or specifications along the way because they cannot meet the original ones?
6. Are the customers usually delighted with these changes in scope?

1.2.1.1 Types of Projects

Table 1.1 illustrates four types of projects. The horizontal axis categorizes projects by absolute deadline versus as soon as possible (ASAP). The vertical axis separates internal projects, generally focused on improving operations, and external projects, generally performed for profit. The answers to the above questions depend on project type. Table 1.1 also lists some examples.

Type I projects are absolute-deadline-driven projects for an external customer.Examples include proposals and major events. Requestors simply do not accept proposals after the specified delivery time. Therefore, proposal teams rarely deliver proposals late. Management usually responds surely and quickly to reward proposal managers who spend the time and money on a proposal and deliver it late. Sometimes, they provide the proposal manager an opportunity to seek employment elsewhere. Likewise, although there may be much adjusting of scope and expediting, other deadline-driven projects usually happen on time. They do not delay the Olympics; they finish the stadium (somehow). People seldom fail to have things ready for a national meeting or prebooked trip. People rarely bow out of elections because their campaign is behind schedule. In these types of projects, usually the money and scope changes, while holding the schedule.

Type II projects do not have specific externally driven end dates (although management may set one internally.) Many projects performed to make money (e.g., new product launch or the construction of a hotel) and most government projects fall into this category. You do not lose all of the benefits because of project delay. You just lose the benefits for some time. (This loss is usually understated or unknown.) In the case of projects that are not end-date driven, all three of the project variables (scope, schedule, time) may change.

Type III and IV projects often compete with each other for funding within a company. Type III projects frequently get higher placement on project priority lists because whatever drives the date often has a penalty associated with overrunning it. Finally, type IV project are the ones that often determine the future of the company. Companies perform type IV projects to improve the company in the future. Therefore, they are always better done sooner. Unfortunately, they often rank lowest in project priority lists, getting starved for resources, and extend on and on.

Regardless of the position of a project in your priority list, I assert “any project worth doing is worth doing fast,”The reason is that project benefits don’t start until

Table 1.1 Four Major Types of Projects Determine How You Should Plan

	Absolute Deadline	*ASAP*
External customer	Type I: Proposal, event, contract with penalties	Type II: Construction
Internal customer	Type III: Y2K, regulatory	Type IV: Product development, process improvement

a project completes, while investment starts at the beginning of a project. Therefore you always increase return on investment by finishing as soon as possible once you start a project.

1.2.1.2 Anecdotal Data

Project management has a long history, reflected in the man-made wonders of the world. However, did they do it on schedule? Did they do it to an approved budget? Did they comply with all specifications and regulations? More and more in recent years, the answers to these questions are no. Most people are aware of major projects that have suffered from the problem; examples include the Denver, Colorado, airport or the Chunnel, connecting England and France, the International Space Station [6], and the Boston Big Dig. Besides being late and overbudget, they also experienced scope problems. The Denver airport baggage system did not work for a long time after the airport opened. The Chunnel had an opening ceremony but could not transport passengers. In 2004 the space station is still mostly sitting on the ground. Many are also aware of the "vaporware" problem in the software industry: almost all software releases later than predicted, full of bugs, and sans many promised features. Microsoft has raised this to a fine art form.

A newspaper article summarized the saga of the Denver airport. It was over two years late. The cost rose from $3 billion to over $4.2 billion. The scope was not complete when it opened. The newspaper wrote the report to give the good news that the airport made a $28 million profit in 1996. Let us see, $28 million on a $4.2 billion investment works out to a return on investment of 0.6% per year. How many investors would put their money in a project like that? Bond investors filed a lawsuit.

Table 1.2 is found throughout the project-management world and is now distributed worldwide across the Internet. It is only one example of many with similar themes, attesting to the fact that projects often fail to achieve success. It is instructive to note that these effects appear to transcend all cultures and national boundaries. Many project-management books include a section why projects fail and offer remedies to the various causes.

1.2.1.3 Quantitative Data

The government is most willing to compile and publish results of quantitative reviews of project performance. Usually, they do not bother to publish good news on contractors, so the published information may be biased. Here are just two quantitative examples:

> GAO [4], following a review of major systems acquisitions (Projects over $75 million) by the United States Department of Energy (DOE) reports:
>
> (1) from 1980 through 1996, DOE conducted 80 projects that it designated as major system acquisitions;
>
> (2) 31 of those projects were terminated prior to completion, after expenditures of over $10 billion;
>
> (3) only 15 of the projects were completed, and most of them were finished behind schedule and with cost overruns;
>
> (4) further, 3 of the 15 projects have not yet been used for their intended purpose;

Table 1.2 The Immutable Laws of Project Management

LAW 1: No major project ever completes on time, within budget, with the same staff that started it, and the project does not do what it is supposed to do. It is highly unlikely that yours will be the first.
- Corollary 1: The benefits will be smaller than initially estimated, if they made estimates at all.
- Corollary 2: The system finally installed will be late, and will not do what it is supposed to do.
- Corollary 3: It will cost more but will be technically successful.

LAW 2: One advantage of fuzzy project objectives is that they let you avoid embarrassment in estimating the corresponding costs.

LAW 3: The effort required correcting a project that is off course increases geometrically with time.
- Corollary 1: The longer you wait the harder it gets.
- Corollary 2: If you wait until the project is completed, it is too late.
- Corollary 3: Do it now regardless of the embarrassment.

LAW 4: Everyone else understands the project purpose statement you wrote differently.
- Corollary 1: If you explain the purpose so clearly that no one could possibly misunderstand, someone will.
- Corollary 2: If you do something that you are sure will meet everyone's approval, someone will not like it.

LAW 5: Measurable benefits are real. Intangible benefits are not measurable, thus intangible benefits are not real.
- Corollary 1: Intangible benefits are real if you can prove that they are real.

LAW 6: Anyone who can work effectively on a project part-time certainly does not have enough to do now.
- Corollary 1: If a boss will not give a worker a full-time job, you shouldn't either.
- Corollary 2: If the project participant has a time conflict, the work given by the full-time boss will not suffer.

LAW 7: The greater the project's technical complexity, the less you need a technician to manage it.
- Corollary 1: Get the best manager you can. The manager will get the technicians.
- Corollary 2: The reverse of corollary 1 is almost never true.

LAW 8: A carelessly planned project will take three times longer to complete than expected. A carefully planned project will only take twice as long.
- Corollary 1: If nothing can possibly go wrong, it will anyway.

LAW 9: When the project is going well, something will go wrong.
- Corollary 1: When things cannot get any worse, they will.
- Corollary 2: When things appear to be going better, you have overlooked something.

LAW 10:Project teams detest weekly progress reporting because it so vividly manifests their lack of progress.

LAW 11: Projects progress rapidly until they are 90 percent complete. Then they remain 90 percent complete forever.

LAW 12: If project content is allowed to change freely, the rate of change will exceed the rate of progress.

LAW 13: If the user does not believe in the system, a parallel system will be developed. Neither system will work very well.

LAW 14: Benefits achieved are a function of the thoroughness of the post-audit check.
- Corollary 1: The prospect of an independent post-audit provides the project team with a powerful incentive to deliver a good system on schedule within budget.

LAW 15: No law is immutable.

(5) the remaining 34 projects are ongoing, many with substantial cost increases and schedule slippage.

Recent updates note, despite sincere efforts to improve project performance [5]. In September 2002, we reported that, based on a comparison of 25 major DOE projects in 1996 with 16 major projects in 2001, it did not appear that DOE's contractors had significantly improved their performance over the period. In both sets of

projects, over half had both schedule delays and cost increases. And the proportion of projects with significant cost increases and schedule delays was actually higher in 2001 than in 1996. For example, 38 percent of the projects we reviewed in 2001 had doubled their initial cost estimates, compared with 28 percent in 1996.

In another report evaluating management of a recent version of the space station by NASA, GAO [5] noted that

> (1) in its June 1997 testimony, GAO noted that the cost and schedule performance under the prime contract had been consistently worsening for some time;
>
> (2) GAO pointed out that between January 1995 and April 1997, the costs associated with the schedule slippage had increased from a value of $43 million to $129 million;
>
> (3) during that same period, the variance between the actual cost to complete specific work and the budget for that work had gone from a cost underrun of $27 million to a cost overrun of $291 million;
>
> (4) as of July 1997, the costs associated with the schedule slippage had increased further to $135 million and the cost overrun had increased to $355 million;
>
> (5) the rate of decline for the cost variance is especially worrisome because it has shown no particular inclination to lessen;

Your tax dollars at work! Consider the fact that these are two different government agencies with very different projects and very different constraints. Yet, performance on both is equally miserable.

The Department of Defense (DOD) shows similar tales of woe. James P. Lewis [7] reports on the cancellation of the A-12 Avenger Program in 1991, causing the loss of 9,000 jobs and entailing a lawsuit by the government for $1.35 billion in contractor overpayments. Lewis notes, "It has been acknowledged by reliable DOD sources that the C/SCSC [cost/schedule control system criteria] management systems were implemented properly, and were functioning well at both the principal contractors." (Note: some would argue that the C/SCSC is the most sophisticated project-management system currently available.)

One (now somewhat dated) study from Australia [8] found that construction projects completed only one eighth of building contracts within the scheduled completion date, and that the average overrun exceeded 40%. Daniel Chun and Mohan Kummaraswamy [9] reported this in a recent study of the causes of time overruns in Hong Kong construction projects. The same study noted, "Delays in construction projects are still very common in most parts of the world, even with the introduction of advanced construction technologies and more effective management techniques."

Software projects seem particularly prone to failure. The most recent quantitative study [10] shows significant progress since 1994:

> Project success rates have increased to just over a third or 34% of all projects. This is a 100% plus improvement over the 16% rate in 1994. Project failures have declined to 15% of all projects, which is less than half the 31% in 1994. Challenged projects account for the remaining 51%.

Project success of one third remains a long way from my standard for success. How about yours?

The only common thread is the project system! They all use the present theory of the CPM. (They may not all use it the same way, and they may not all use it well, but nearly all at least claim to use it.)

There are several precursor conditions that you should satisfy before starting any project. You can make improving project management a project. The same necessary conditions apply. You need to do the following:

- Be sure you are working on the correct problem (right problem).
- Assure that the overall objective of the project, when achieved, solves the correct problem with an implementable solution (right solution).
- Develop a scope and design that delivers the solution.
- Execute the project to deliver the designed scope, achieving the objective within the planned time and budget (right implementation).

The last point reiterates the three necessary conditions for any project.

1.2.2 But Some Companies Make a Lot of Money Running Projects

Despite the gloom-and-doom reports above, many companies prosper in the business of running projects. What do these companies do that the losers aren't doing? Much of the project literature would lead you to believe that they are the precious few who follow the Project Management Body of Knowledge (PMBOK™) in the most detail and that all you have to do to join the successful is more of what you are doing, and do it faster.

Successful project-management companies have put in place systems that allow them to win in their environment. This environment generally includes competitors using a similar system. A competitive system does not require you to be great or even good. It does not require that your theories be right. To do well, you just have to be a little bit better than your competitors are. Often you can maintain this improvement through operational excellence, even with a system that has fundamental weaknesses. However, overcoming the fundamental system weakness provides the opportunity to steal the market if the improvement is not easily, or at least rapidly, matched by the competitors.

The present project systems must also allow some of the people in the company to win, as they need people experienced in their system to make it work. I rarely hear about the potential impact on the rest of the people in the company or of how their suppliers get along. The model we develop of present performance predicts significant expediting, exploiting, and stress among the project participants.

One feature seems common to the project systems of successful project companies. The PMBOK™ Guide considers it. Authors sometimes mention the lack of it among the reasons projects fail, but perhaps not often enough. The answer is every company that succeeds in the project-management business uses an effective change-control process. This process allows them to account for changes that happen to the project along the way and to recoup any financial impact from such changes. Many of the students that attend the project-management classes I teach complain about "scope creep." I tell them my projects never experience scope creep and that I consider scope creep to be a self-inflicted wound for a project manager. Successful project managers always control scope. Scope control is a primary job

function for a project manager. I tell the (sometimes wide-eyed) students that I love proposed changes. But, I control them; assuring the requestor I will implement them immediately after they are approved by the project customer (even if it is the customer that is "directing" a change). I then solicit that approval after rigorously estimating the scope, cost, schedule, and risk impact of each change, including the impact of cumulative small changes. It is amazing how the frequency of scope change requests reduces when you are serious about this.

An effective change process is one way to handle variation while applying the present system. Subsequent chapters reveal why it is not the best way to handle some project-performance variation. An effective change-control process is a necessary part of an effective project system. The critical-chain method requires effective change control, but dramatically reduces the number of changes.

1.2.3 Problem Cause, or Better Defining the Problem

Defining the problem at a high level is easy. Project managers must meet customer needs on time, at or under budget, all of the time. Evidence presented above demonstrates that the present theory does not produce this desired result. The problem is to invent a better theory that does produce the desired effect (DE).

The Avraham Y. Goldratt Institute asks project-management students why it is so difficult to meet the three necessary conditions for a successful project? The usual answers include things like

- Unforeseeable bad weather;
- Unforeseeable difficulties at vendors who supply equipment;
- Longer-than-expected time in meeting government requirements;
- Unrealistic schedule;
- Unreliable (but cheaper) vendors or contractors;
- Difficulties in matching operators available with project needs;
- Unforeseen emergencies.

And so on. The lists usually have two things in common: whatever causes the problem is outside the control of the project manager, and the cause is some type of unexpected event.

Many project-management texts include lists of the reasons projects fail. One remarkable aspect of these lists is that they list different things. Some of the lists compare the reasons for project failure viewed by different people (e.g., viewed by the project manager and upper management). These lists disagree on the importance of various causes. A second remarkable aspect of these lists is that none of them suspects the project system. Two assumptions underlie many of the evaluations leading to these lists:

1. *Project work is deterministic*. The evaluations address reality as if it is possible to get "accurate" or "precise" single-point estimates. Therefore, they assume variation in the result must be caused by failure to define or operate effectively.

2. *The present project-management system is effective.* This assumption leads to solutions that identify the particular part of the existing system that did not function well, causing a particular failure. None of these studies questions the effectiveness of the assumed system (which is often poorly defined in the studies themselves). None of these studies questions the assumptions underlying the assumed effective system.

One way to begin to understand project success or failure better is to look at the system and understand some of the assumptions that underlie it. Following Aldo Leopold [11] (who was working in an entirely different problem domain), we can identify factors and influences that affect the success of projects. Factors are things that more-or-less directly affect project success in terms of the three necessary conditions. Success factors include

1. Selection of the right problem;
2. Selection of the right solution;
3. Creation of a satisfactory plan;
4. Use of an effective project-control system;
5. Effective project execution;
6. Use of an effective method to manage uncertainty.

Further expansion of item 4, an effective project-control system identifies:

4.1 Resource quantity;
4.2 Resource skill;
4.3 Resource behavior;
4.4 The project-management process;
4.5 Project execution tools;
4.6 Project changes.

While this list of factors is certainly not complete, it captures many of the items addressed in project-failure studies.

In addition to the factors that seem to directly influence project success, you can also identify items that influence these factors. Project-success influences internal to the project team may include

1. Management;
2. Measurement;
3. Rewards;
4. Policies;
5. Social norms;
6. Variation in the processes that produce project results.

Influences external to the project team may include

1. Competitors;
2. Suppliers;

3. Client;
4. Regulators;
5. The physical environment;
6. Other stakeholders (e.g., the public).

Influences may affect one or more of the factors that more directly affect project success. Table 1.3 illustrates the relationship between the influences and the factors, as well as the author's indication of the stronger influences.

Note that the factors are not independent of each other. Likewise, the influences are not necessarily independent of each other. Thus, there are relationships between all of the variables. The project performance system is a complex system indeed. This, combined with the sheer number of factors and influences, may explain why people attribute project failures to such a wide range of causes.

System theory, described in Chapter 2, clarifies that influences (i.e., relationships between the factors) can be more important than the factors themselves when we seek to improve a system. Reasons for this include the fact that the influences may affect many factors and that the influences may be more subject to direct intervention (change) than the factors. This is certainly true for management-controlled influences, such as the measurement and reward systems, and the policies of the company.

Table 1.3 Factors and Influences Affecting Project Success

	Influences on Project Success Factors								
	Internal					*External*			
	Management	Measurement	Rewards	Policies	Social	Competitors	Suppliers	Client	Regulators
Factors That Determine Project Success									
Right Problem	X				X			X	O
Right Problem	X				X			X	
Effective Plan	X	X		X				O	O
Project Control System	X	X		X	X			O	O
Project Execution									
Resource quantity	X			X		O	O		
Resource skill	X	X	X	X	X		O		
Resource behavior	X						O	O	
Work processes	X						O		
Changes	X		X				O		
Uncertainty	X		O				X	O	X
	X	Significant influence							
	O	Some influence							

The problem statement that Goldratt proposed in order to develop the critical-chain method blamed poor project performance on the system. He asked, "What is it about the current system that causes so many projects to fail?" He had a good hint from his previous work with production systems and theorized that the project systems failed to manage uncertainty effectively.

1.2.4 Right Solution

People have posed many solutions to improve project management over the last forty years. They attempt to better meet customer needs on time, at or under budget. Solutions generally tend to provide more and more detail in the planning, measurement, and control of the project. Improved availability of PC-based project-management systems leads to defining more tasks on projects. The software helps to automatically create a project network, define a critical path, allocate resources, and measure project performance at any level of detail.

Goldratt begins *Critical Chain* [12] with a discussion of a company wanting to reduce the time on critical development projects. They had an extensive analysis performed by expert consultants, who looked at their project-management system and recommended many changes. In discussing the amount of time saved by making all of these changes, they conclude that they would save "Maybe five percent. Maybe not even that."

1.2.4.1 Do More Better

Earned-value and derivative cost-/schedule-control systems (CSCS, or CS^2) [13] frequently increase the detail of project plans and measures. The procedures that companies put in place for people to use these systems often are many hundreds of pages long. The number of activities in project schedules reach the tens of thousands. They sometimes force activity durations to short times, such as "no more than two weeks."

The author worked with one government agency that followed the process of requiring increasingly detailed planning over a period of twenty years. Each time they had a project problem, they blamed some people, investigated the cause of the problem, and put in more procedures. The minimum time to do a project crept up to almost seven years, not including the time to do the project! That is, they built in seven years of planning time before the start of any project. There were engineering studies and conceptual design reports and independent cost estimates and validated cost schedule control systems. Yet, the cost and schedule of projects continued to rise, and more and more projects failed to meet technical requirements. In one case, they canceled a project after spending over a billion dollars on it. Other projects are tens of years late.

One study showed that it cost them four times as much per square foot as local construction by nongovernment purchasers to build a simple office building. Projects were having larger and larger crises, where they would "rebaseline," yielding new cost and schedule estimates several times (usually three or more times) larger than the original estimates. They cancelled larger and larger projects because the need was gone before the project was over or because the newly projected cost and schedule estimates changed the cost-benefit equation to where the project no longer

made sense. This is the problem they were trying to solve in the first place. Is the world changing that much? On the other hand, could it be that our solutions are actually making things worse, not better?

Let's review the logic of the "do more better" approach. If your objective is to reliably complete projects to scope, schedule, and cost, you must define those requirements accurately. To define requirements accurately, you must add detail to your project plans, because previous projects failed to deliver at the present level of detail. This logic seems to make sense and is in line with literature that attributes project failure to inadequate requirements or insufficient detail in the project plan.

The "do more better" approach frequently leads to project plans with tens of thousands of activities. We recently worked with a client who was rather proud of the fact that his project plan contained over 30,000 activities.

Later on, I will go into the factors that will help you determine the size of activities you should have in your project. In some cases, this may lead to thousands of activities for very large projects. But, to put it in perspective, consider a much more modest project plan containing a mere 100 activities. The average size of an activity (measured in dollars, or person-days, or even in task-days) is, by simple math, 1% of the total project (by comparable measure.) Most project managers would be very happy to have their project come in within 1% of plan. The problem with project success must involve something that causes variations of far more than 1%. Therefore, it seems to me that simply increasing plan detail beyond 100 activities is not going to improve project success. Something else must be at work.

Sometimes people defend the more-detail method by suggesting that the problem, even though it is much bigger than 1%, is that they have missed something in their plan. You are not likely to find the missing 20% inside the 1% chunks of the project. Looking inside the 1% for the big hitters reminds me of the story about the drunk who lost his car keys around the corner but is looking for them under the street light because he can't see anything over there in the dark. If you are worried about missing big chunks, you are far better examining the spaces between the 100 activities that you have, rather than breaking the defined activities into greater and greater detail.

Some of the literature that poses causes and solutions to project problems also offers anecdotal evidence that a given solution worked to improve project success in one or more subsequent projects. While this evidence is interesting, it does little to prove that the writers have really solved the cause of lack of success in project systems. Reasons for this include the following:

1. Theory of knowledge: One or more successful cases do not prove a theory (discussed in Chapter 2).
2. Environment: Their environment may have had very poor practices to begin with and been one in which any degree of discipline is likely to cause an improvement.
3. Regression to the mean: A particularly bad performance is likely to be followed by a better performance.
4. The Hawthorne effect: This is a psychological effect whereby workers singled out to try out new methods respond positively to any change,

including changing back to the conditions that existed before the experiment.

In other words, the posed theories have not been subject to effective critical thinking and experimental tests.

1.2.4.2 Variation and Uncertainty

Everyone knows that project tasks have a certain amount of inherent variation and frequently uncertainty about the amount of that variation. The very definition of a project says you have not done this task before; it is unique. At least, you have not performed all of the tasks the same way you will in this particular project. In order to complete the project successfully, you must account for this variation and uncertainty. People's ability to estimate accurately depends on a number of factors. There is substantial evidence to indicate that people tend toward overconfidence in their belief in the accuracy of their estimates [14]. It is unlikely that most project tasks can be estimated more accurately than ±20%.

We have people estimate a very simple task as part of our training classes. Nearly all of the participants in the exercise agree that the task is much simpler than most of their project tasks. They also agree that the ability of the other people in the room to estimate this task should be as good as, or better than, the ability of their project estimators to estimate project estimates. The range of the estimates usually is several hundred percent of the mean. The standard deviation is usually on the order of 30% of the mean. Figure 1.4 illustrates combined results from several of these exercises.

Figure 1.5 illustrates the expected general behavior of the accuracy of a single task estimate as a function of the amount of effort put into creating the estimate. The accuracy scale presents the accuracy as a percentage of the mean estimate, so a perfectly accurate estimate has an uncertainty of zero. An estimate with no effort at

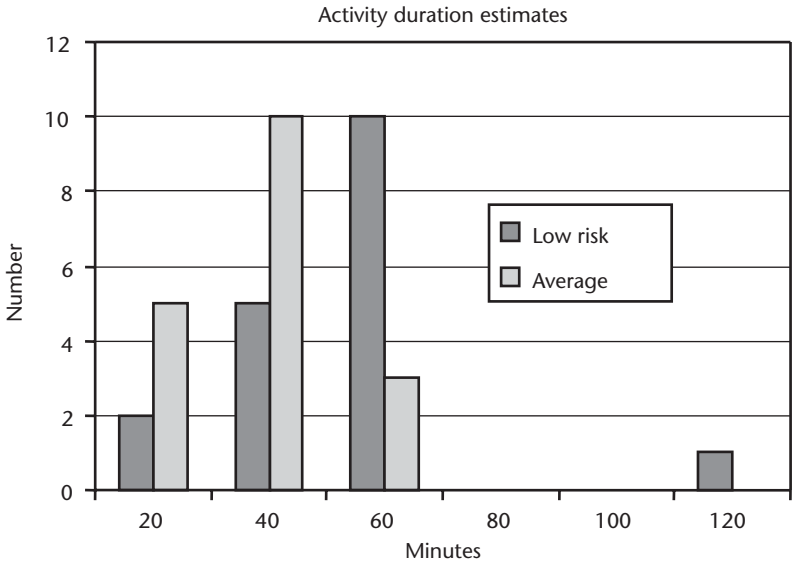


Figure 1.4 Estimate uncertainty for a very simple project task illustrates the typical range of real uncertainty.

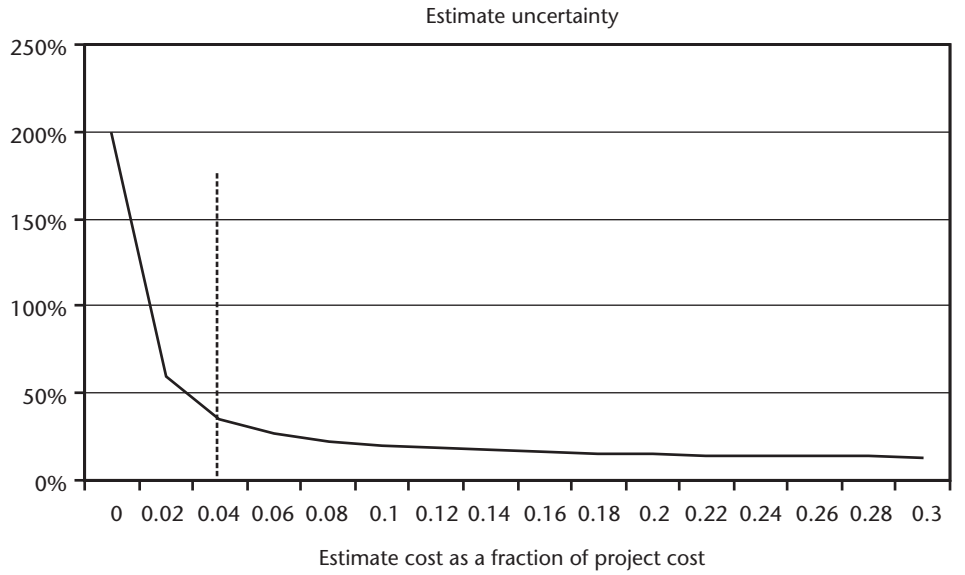


Figure 1.5 Estimated accuracy generally increases with the effort applied to the estimate, up to a limit determined by the process involving the subject of the estimate.

all should have an accuracy of at least 100% on the down side and could be orders of magnitude (hundreds of percentage points) too low. The curve illustrates that the accuracy should generally improve as more effort is put toward the estimate. A lower limit usually limits improvement due to the inherent variation in the process that will produce the task result. This lower limit, described further in Chapter 2, is called common-cause variation. No matter how much more effort you put into the estimate, you can *never* do better than the common-cause variation of the process that produces the result of the task. You can only reduce common-cause variation by changing the task process.

Consider two regions of Figure 1.5, divided by the vertical dotted line. To the right of the line, adding more effort to the estimate does not significantly improve the accuracy of the estimate. If you are far to the right of the line, reducing the effort should not have much impact on uncertainty. Estimates to the left of the line show increasing sensitivity to the amount of effort applied. Small reductions in the applied effort will greatly increase the uncertainty of the estimate. Small increases in the effort will significantly improve the estimate.

Assuming that the tasks you have identified in your project plan identify deliverables (clean handoffs from one primary resource type to the next), the effect on overall plan uncertainty that will obtain from subdividing tasks to your project plan depends on the region in which you are operating. With a fixed level of investment in the estimate, if you are well to the right of the line, adding more tasks (which reduces the effort per task) may increase the accuracy of the overall plan. The reason is that the accuracy of the overall plan improves as the plan is divided into more equal-sized pieces, if the accuracy of the individual pieces is the same.

If the amount of estimating effort you can afford puts you near, or to the left of, the vertical line in Figure 1.5, adding more tasks to your plan can decrease the accuracy of the overall plan. The reason is that the increasing uncertainty in each task estimate can be much greater than the statistical benefit of more individual tasks.

Adding more tasks to a project plan increases the number of potential task connections much faster than the number of tasks you add. For example, if you add 1 task to a plan with 100 tasks, you only add 1 task. You add the potential for 200 additional connections, however, as each task in the existing plan is a potential predecessor and a potential successor to the task you just added. The additional potential relationships greatly increase the probability of errors in the project-task network as the number of tasks in the plan increases.

From the alleged causes of project failure posed so far, you might deduce that uncertainty causes projects to fail. If this were the case, you should predict that all projects with uncertainty will fail. Based on the definition of a project and our understanding of the real world, all projects have uncertainty. Therefore, you might predict that all projects would fail. Many do, but not all. Furthermore, there is evidence that some projects succeed despite extreme uncertainty. Goldratt describes one airplane project that defies this prediction in *Critical Chain*. The designers developed an airplane with unprecedented capabilities in eight months instead of the ten years such developments normally take. There are other cases. The United States did succeed to meet President Kennedy's objective to put a man on the moon by the end of the decade. The moon project was one of the most uncertain projects ever undertaken by man. Creation of the atomic bomb was a similarly uncertain project completed in a remarkably short time.

There have been substantial efforts to reduce the uncertainty in project estimates and the variation in performing project tasks. There are excellent tools for estimating projects and project tasks that no doubt help improve the accuracy of estimates and, more importantly, collect data to estimate the variation in project tasks. There have been improvements in performing project tasks, using approaches such as Six Sigma. Unfortunately, the project-failure data includes companies that have applied such techniques to minimize variation. Collectively, they have not made much difference.

A cornerstone of the scientific method is that scientists can never prove that any scientific theory or law will continue to work in the future, but they can disprove a theory with just one proper test. More than one instance proves that uncertainty itself cannot be the cause of project failure.

If simple uncertainty does not meet the test of explaining project failure, can you modify the theory to fit the known evidence? You know that some projects use different ways to manage uncertainty. For example, the Apollo project managed risk by hiring three companies to produce three different solutions for high-risk developments. They chose one as the primary path, but had two backups in case the primary path failed. They planned on much test and retest. (And they had plenty of spectacular failures along the way.) While this is an expensive way to manage uncertainty, it worked. Goldratt used thinking like the above to pose the hypothesis that it is *failure to effectively manage uncertainty* that causes most project failures. Chapter 3 examines this hypothesis in depth. If he is right, the direction of the solution is to create a different project system more able to manage uncertainty.

1.2.5 Right Execution

Right execution refers to execution of the solution to the problem. Improvement to the project system is a project.

Goldratt noted in his pamphlet "My Saga to Improve Production" [15],

> It took me some time to figure it out, but at last I couldn't escape the simple explanation: the efforts to install the software distracted the plant people from concentrating on the required changes—the changes in fundamental concepts, measurements and procedures.

A similar phenomenon occurs in many efforts to improve the performance of the project system, and I find I have to fight it almost every time a company chooses to implement CCPM. Many people quickly focus on the software as the solution. Software alone never is the solution, and if they focus on the software, they usually do not get much benefit. In other cases, the usual solutions are along the line of doing the present system better, which many people interpret as requiring more detail and more documentation. This often involves installing new project or database software. These solutions distract people further from performing the project. Such efforts seldom seem to improve much. Of course, better implementation of a flawed system is unlikely to improve much anyway. Chapter 10 provides an effective plan to implement the CCPM system.

1.3 Success with Critical-Chain Project Management

Having defined the problem and substantiated the claim that the present theory is in need of improvement, the next step requires creating a new theory (of the project system): CCPM. Expectations for this theory are that it will, subject to critical evaluation, demonstrate greatly enhanced and consistent success in achieving project success. (We are looking for a 50% improvement, not 5%!) It should explain both past success and failure and provide testable predictions of future performance. Growing experience with the new theory shows benefits that far exceed the minimal performance requirements for the new theory but that the theory can explain. These benefits (compared to the present critical-path theory) are

- Improved project success:
  - Projects complete on time all the time;
  - Projects deliver full scope;
  - Projects finish under budget;
  - Projects result in improved market position and business growth.
- Reduced project duration:
  - Projects complete in one half the time (or less) of previous similar projects;
  - Individual project plans reduce by at least 25%;
  - Multiple project durations reduce by larger amounts;
  - There are reduced project changes;
  - There are early returns for commercial projects;
  - There are reduced payback periods for investment projects.
- Increased project team satisfaction:

- There is reduced confusion from multitasking;
- There is ability to focus on one task at a time;
- There are reduced changes;
- There is reduced rework;
- There is reduced pressure from multiple project managers;
- Win–lose task completion (date-driven task pressure) is eliminated;
- Individuals use the buffer report to decide their own task priorities;
- There is reduced insertion of new priority tasks;
- Project measurement is simplified;
- Plan status is quick and easy;
- Real-time project status is available, so there's no need to wait for financial reports;
- Status provides immediate focus by buffer, chain, and task;
- The buffer report defines decisions;
- Buffer reporting focuses decisions on management priorities (reflected in the buffers by staggering project starts).

- Simplified project management:
  - There is clear focus for project manager (critical chain, reduced early start);
  - Simplified project plans reduce paperwork;
  - Project status reporting is simplified;
  - Measurement decides whether to plan or act;
  - Measurement decides resource priorities.
- Increased project throughput with same resource:
  - Resource-demand conflicts are reduced;
  - More projects complete faster for the same level of resources;
  - There is less need to hire new critical resources;
  - There is less delay due to resources;
  - Project cash flow is improved;
  - Return on investment is improved.

Evidence of other users often gives people confidence to try new ideas. The present CPM project paradigm has been in force for over forty years, making change very hard for many people to accept. More and more companies, small and large, are demonstrating success with CCPM. Several examples illustrate this success. (As mentioned earlier, success examples do not prove a new theory. They only provide confidence that it is not fatally flawed.)

1.4 Honeywell DAS [16]

> The RNLAF team was asked by the customer to deliver something we originally scheduled to take 13 months to deliver—and the team did it in six months…the

team is experimenting with a new way of scheduling the program using critical chain concepts. Boeing has read the book, and is supporting the concept.

1.5 Lucent Technologies [17]

Lucent Technologies has adopted CCPM as their primary tool for project management. (The author provided Lucent training and implementation assistance.)

> In 1996, Lucent Technologies Advanced Technology Systems, now part of General Dynamics, was told by a sister organization that the yearlong project being considered was an impossibility…the project was used as a pilot effort, to evaluate TOC project management. The project was completed in June, 1997, with buffer to spare.

1.6 Israeli Aircraft Industry

The Israeli aircraft industry employs about 15,000 people. A major function is to maintain jumbo jets used in passenger service. A particular type of maintenance, called "type D," normally takes 46 days in the industry. The penalty for nonperformance to schedule is very steep, $60,000 per day, because the airlines need the planes back into scheduled service. The company had been paying up to $25 million per year in penalties. A letter from the manager to Goldratt [18] notes, "[W]e succeeded to drop our average Turn Around Time per Aircraft Visit from three months to two weeks and to increase our backlog from two months to one year."

1.7 U.S. Navy Shipyards

The U.S. Navy has been implementing CCPM at a number of naval shipyards. One of the greatest successes was with the 2001 maintenance of the U.S.S. *Harry S. Truman*, one of the world's largest ships. Adopting selected TOC and CCPM behaviors alone (i.e., using existing legacy project software) enabled the project team to deliver this very large project early and save over $20 million. Subsequent application at the Pearl Harbor Naval Shipyard yielded a schedule performance increase of from 40% to over 90% and a productivity increase of about 100% on more modest-sized projects performing maintenance on U.S. nuclear submarines. The U.S. Navy is in the process of extending the use of CCPM to more and larger projects at the four public shipyards and plans to extend the use to the private shipyards that support them.

1.8 Summary

This chapter has defined the problem that this book aims to resolve and identified CCPM as the new theory (hypothesis) to resolve the problem. Key points are

- Project success rates using the existing critical-path paradigm, or CPM, is improving but remains poor for all types of projects in all types of cultures.

- Hypothesized causes of project failure do not address potential causes outside of the existing project system, most often leading to remedies that work harder with the old system: the "do more better" approach. This does not seem to address the right problem.
- Improvements using the "do more better" approach and efforts to reduce variances in estimates or performance of individual project tasks show a low return on investment (e.g., returns on the order of 5%), with large effort invested.
- Growing evidence supports the assertion that the right problem is in the design of the project system itself, specifically that the system fails to properly manage the reality of uncertainty.
- The right solution requires a project system that has a much higher success rate and that is simple to use.
- A growing body of evidence does not contradict the hypothesis that Goldratt's critical-chain method satisfies the necessary conditions for project success, causing improvements on the order of 50% or more, with relatively small investment.

Comparing the results of applying the critical-chain theory to the existing theory (i.e., the critical-path theory as described in the PMBOK™ Guide) provides support for using the CCPM theory while we continue to review and improve it.

References

[1] PMI, *A Guide to the Project Management Body of Knowledge*, Upper Darby, PA: PMI, 2000.

[2] Goldratt, Eliyahu M., *The Goal*, Great Barrington, MA: North River Press, 1984.

[3] Popper, Karl R., *Objective Knowledge, An Evolutionary Approach*, Oxford: Clarendon Press, 1997, p. 144.

[4] GAO/T-RCED-97-92, "Department of Energy: Improving Management of Major System Acquisitions," Testimony, March 6, 1997.

[5] GAO-03-570T. *Status of Contract and Project Management Reforms. Statement of Robin M. Nazzaro, Director Natural Resources and Environment*. March 20, 2003.

[6] GAO/T-NSIAD-97-262, "Space Station: Deteriorating Cost and Schedule Performance under the Prime Contract," Testimony, September 18, 1997.

[7] Lewis, James P., *The Project Manager's Desk Reference,* Chicago: Irwin, 1995, p. 245.

[8] Bromilow, F. J., "Measurement of Scheduling of Construction Time and Cost Performance in the Building Industry," *The Chartered Builder,* Vol. 10, 1974.

[9] Chun, Daniel W. M., and Mohan M. Kummaraswamy, "A Comparative Study of Causes of Time Overruns in Hong Kong Construction Projects," S)263-7863(96)0039-7, *International Journal of Project Management*, Vol. 15, No. 1, February 1997.

[10] Standish Group, "Latest Standish Group CHAOS Report Shows Project Success Rates Have Improved by 50%," available at http://www.standishgroup.com/press/article.php?id=2 (accessed April 29, 2004).

[11] Leopold, Aldo, *Game Management*, Madison: University of Wisconsin Press, 1933.

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

[13] Lambert, L. R. "Cost/Schedule Control Criteria (C/SCSC): An Integrated Project Management Approach Using Earned Value Techniques," *The AMA Handbook of Project Management*, New York: AMACOM, 1993.

[14] Kahneman, Daniel, Paul Dlovic, and Amos Tvershky, *Judgment under Uncertainty: Heuristics and Biases*, Cambridge: Cambridge University Press, 1982.

[15] Goldratt, Eliyahu, "My Saga to Improve Production," New Haven, CT: Avraham Y. Goldratt Institute, 1994.

[16] Honeywell Defense Avionics Systems, Albuquerque, New Mexico, *Horizons,* Vol. 5, No. 2, February 20, 1998.

[17] Rizzo, Anthony, "The TOC Solution of R&D and Multi-Projects Organizations," Lucent Technologies, Whippany, New Jersey, January 5, 1998.

[18] See http://www.Goldratt.com (Web site of the Avraham Goldratt Institute).

CHAPTER 2

TOC, PMBOK™, Lean and Six Sigma

This book approaches the problem of improving project management from the perspective of synthesizing two domains of knowledge: the PMBOK™ [1] and the TOC. We consider this synthesis with perspectives from two other knowledge areas: Lean Manufacturing and Six Sigma. This chapter addresses each of these knowledge areas in order and illustrates their relationship to the overall CCPM approach.

These knowledge areas provide different reality filters, or paradigms, to understand the project system. Multiple perspectives enable deeper understanding of the theory underlying CCPM, which I define as the synthesis of Dr. Eliyahu Goldratt's critical-chain approach to schedules and the rest of the PMBOK™. The underlying theory enables you to deal with issues unique to your environment or project.

Figure 2.1 illustrates how the multiple perspectives on the project system might look at problems in project performance. The PMBOK™ perspective compares actual project system performance to the PMBOK™ model, which it assumes to be correct. Therefore, the PMBOK™ perspective is unlikely to blame elements of the PMBOK™ project system as the cause of the problems. It is much more likely to

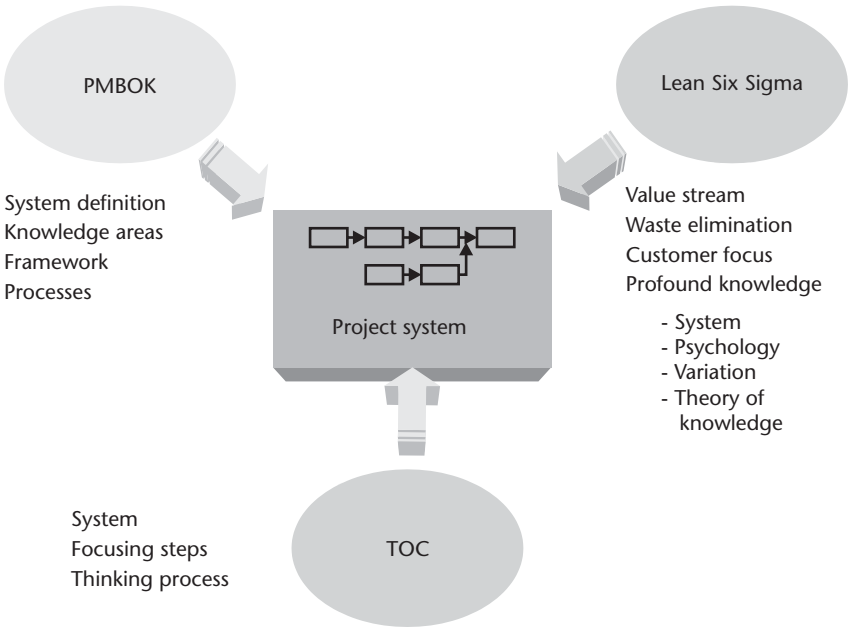


Figure 2.1 Multiple knowledge areas increase perspective on the project system.

blame performance problems on failure to execute properly the (assumed) effective system. This is indeed the nature of much of the project-management literature, as described above. Dr. W. Edwards Deming has noted that you should not expect significant system changes to come from within the system. A natural consequence of solutions based on the PMBOK™ perspective is to "do more better."

Some people have fed back to me a misperception of my view of the PMBOK™ Guide, based in part on the previous paragraph. They asserted that I must not support the PMBOK™ Guide and/or all of the supporting literature. This view is incorrect. I believe that the PMBOK™ Guide represents the combined best knowledge of how to execute projects effectively, and I strongly encourage project managers to become expert in its use, including becoming project-management professionals (PMPs). I strongly support continuous improvement of the PMBOK™ Guide and have contributed to the last two versions. I view this book as part of my ongoing effort to improve project-management systems, and I expect the PMBOK™ Guide to embody some of the methods I describe here as they become more common. I also believe many of the elements of the PMBOK™ Guide are necessary conditions to successfully deploying CCPM and will identify them in the appropriate places.

Six Sigma and its predecessor, Total Quality Management (TQM), seek to continually improve every process, the later through projects that demonstrate a return on investment. These perspectives therefore tacitly assume that the best way to improve a system is to improve every process. A leading consideration in TQM (profound knowledge) provides four subperspectives leading to deeper understanding of the potential causes of project problems. TQM provides specific tools to perform root-cause analysis to identify the causes of problems and develops strategies to remove these causes.

The TOC perspective identifies the system constraint and works to improve its throughput. It provides a system view of projects and a specific theory to predict project performance and the impact of changes to the system. This perspective differs from the PMBOK™ view by considering the project system as a dynamic process to create completed projects. TOC looks at individual project tasks as the operation of a system for producing the result or output of the tasks. It focuses on the fact that the task-performance process includes natural variation and that individual project tasks interrelate.

2.1 Project Management Body of Knowledge (PMBOK™)

Project management made a great leap forward in the 1950s and 1960s with the advent of the CPM and the Program Evaluation and Review Technique (PERT). PERT was developed in 1958 as a joint effort between the United States Navy and the Booz, Allen, Hamilton consulting firm for the Polaris submarine project. These methods were enabled by the advent of computers and were successful in managing the Apollo project to put people on the moon (surely one of mankind's finest hours), as well as many large defense projects.

PCs have brought sophisticated computer-scheduling techniques to everyone's desk. Cost-schedule control systems have increased the complexity of these systems. However, there has been little progress in improving the success rate of projects and

less innovation in the underlying basis and system. People continue to work with project-management assumptions conceived forty years ago.

Figure 2.2 illustrates the related knowledge areas identified in the PMBOK™ Guide. This text focuses on and proposes changes to the project-management knowledge elements to impact the necessary conditions for project success. These are project integration management, project scope management, project time management, and project-risk management. You must address the other knowledge areas to varying degrees, depending on your projects and the environment in which you work.

The PMBOK™ Guide describes general processes for each of the knowledge area, collected into five types of processes:

1. Initiating;
2. Planning;
3. Controlling;
4. Executing;
5. Closing.

These process phases roughly correspond to phases of projects, but there is considerable overlap. The PMBOK™ Guide emphasizes that there are relationships and interactions between most of the project system processes.

2.1.1 Project Integration Management

Project integration management includes project plan development and execution and overall change control through the life of the project.

2.1.2 Project Scope Management

Project scope management includes the process leading to the initiation of the project and scope planning, definition, verification, and change control. Primary

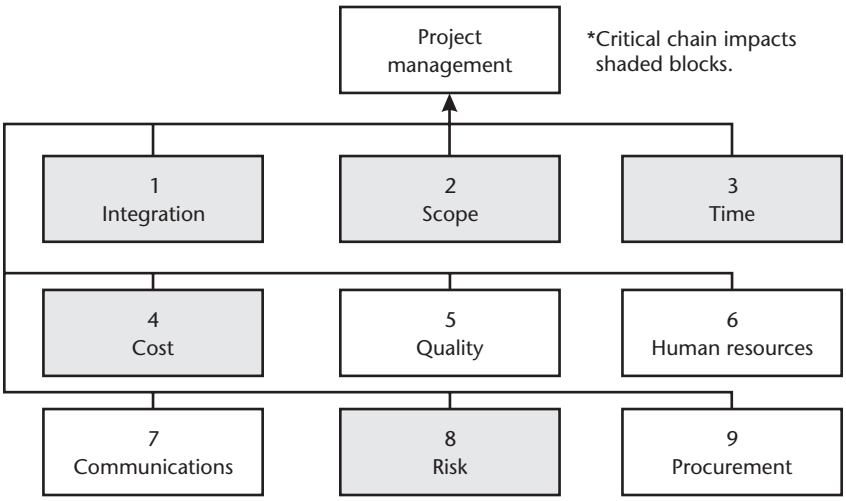


Figure 2.2 The PMBOK™ areas identify the project system.

outputs of the scope-management processes include a project charter, the project work breakdown structure (WBS), detailed statements of work (SOWs), functional and operational requirements (F&OR) or other definitions of the deliverable scope, the project assumptions, and a process for scope change control.

Project assumptions assist planners to develop a deterministic project plan. The plan and control processes defined by the PMBOK™ do not include a way to handle decision branches in a project plan. Assumptions define uncertainty sufficiently to permit defining a deterministic scope, cost, and schedule.

2.1.3 Project Time Management

Project time management includes defining the activities necessary to produce the project scope, sequencing the activities, estimating activity duration, developing the project schedule, and keeping the project on schedule. Schedule preparation requires the WBS and scope statements as inputs. The schedule-development process identifies the activity resource requirements and other potential project constraints. The PMBOK™ Guide notes that activity duration estimates should specify uncertainty and refers the reader to discussions on project-risk management to handle this uncertainty. The guide also discusses the need to level resources in the plan. It does not differentiate between common-cause variation and special-cause variation (see 2.5.2).

The PMBOK™ Guide addresses cost management as a separate topic from time management, but the processes are identical. The schedule-and-cost-control process includes updating the project schedule and budget estimate, planning and executing corrective action, and assessing the lessons learned at the close of the project.

2.1.4 Project-Risk Management

Project-risk management includes identifying and quantifying risks and planning and controlling response to risk. Risk includes both the likelihood and consequences of adverse impacts to the project. The PMBOK™ Guide does not distinguish between common-cause and special-cause variation (see 2.5.2) but appears to lump them together in the performance of risk management. (By the way, I have never heard Goldratt use this distinction either and find that he and many of his followers lump them together under variation, or "Murphy." For reasons I will clarify later, I feel that dumbing-down the understanding of uncertainty in this way is a mistake.)

2.1.5 Other PMBOK™ Areas

The other PMBOK™ knowledge areas, including quality, human resources, communications, and procurement management, are all important, in varying degrees, to projects. They are important to any type of business. In the interest of focus, however, this text will not cover these areas.

2.1.6 Organizational Project Maturity Model

The PMI published the Organizational Project Maturity Model (OPM3) [2] in 2003. The purpose is to "describe a Standard for organizational project management and organizational project management maturity." It appears to me to follow the

Software Engineering Institute's (SEI) Capability Maturity Model™ (CMM™) [3]. The team developing OMP3 reviewed 27 alternative maturity models and assigned teams of three to assess 17 of them in greater depth before designing the OPM3 model. OPM3 contains exhaustive checklists to evaluate an organization against characteristics of high-performance project organizations.

OPM3 takes a significant step beyond the PMBOK™ Guide by including the entire organizational system, as compared to the elements necessary for a single project. It extends the domain of the process groups in the PMBOK™ Guide to the domains of program management and portfolio management, each comprising five process groups. The process groups identify core and facilitating processes. The process-group development does not explicitly consider TOC and CCPM.

It is too early to judge the utility of OPM3. My experience applying the SEI CMM™ to develop a project-management office in an information-technology organization led me to believe that models have utility when focused on the elements that make a difference to the organization's current constraints. One must be cautious to not get lost in the complexity of detail to which such an approach might lead. I suspect one can use OPM3 the same way (i.e., to help apply the TOC focusing steps).

The development of OPM3 revealed an interesting aspect that may relate more to the topic of this book than OPM3 itself. OPM3 notes:

> The strategy, up to this point in Q1 2000, had reflected largely a classic "waterfall" development approach: initial research was to feed into design, design into build and test, and so on. But, there were difficulties associated with the analysis of the qualitative research, and PMI asked the team to do everything possible to accelerate the project timetable.
>
> The OPM3 Guidance Team modified its strategy in two ways: to move away from the "waterfall" development model towards a strategy that aligns more to "rapid prototype development." (p. 55)

Although the PMBOK™ Guide addresses alternative development cycles, including the spiral development cycle inferred above, many organizations struggle with the problem of incomplete or unknown requirements at the outset of projects. I learned long ago to apply the "rolling-wave" planning approach in such situations. The rolling-wave approach develops plans as far ahead as you can develop a realistic known scope and includes tasks to replan further as your project develops the new information to do so. Section 2.3 describes this in a little more detail.

2.2 Lean

J. Womack, D. Jones, and D. Roos introduced the world to Lean thinking with *The Machine That Changed the World* [4]. They defined the principles of Lean production to include

- Teamwork;
- Communication;

- Efficient use of resources and elimination of waste;
- Continuous improvement.

Womack and Jones [5] expand on these principles to emphasize the Lean focus on waste:

- Specify value.
- Identify the value stream.
- Focus on flow of work.
- Implement customer pull.
- Strive for perfection.

These principles align very nicely with TOC by simply aligning value with the company goal. They also align with Six Sigma, but they put more emphasis on the system by focusing on the value stream and emphasize the ideas of customer pull and flow in a way that differs from Six Sigma. The U.S. Navy has defined their synthesis of Lean and Six Sigma as Lean Sigma.

Lean approaches have been finding their way into the world of project management with a delay similar to that experienced by TOC, perhaps in part for one of the same reasons: categorizing Lean as a production approach versus a project-management approach.

W. Dettmer [6] provides an excellent comparison and contrast of TOC and Lean approaches, concluding, "TOC provides a useful system-level framework for directing lean thinking efforts where they will do the most good (the system constraint) and avoiding the pitfalls of applying them where they will do harm."

Dettmer also identifies some significant advantages to synthesizing Lean and TOC, including the following Lean tools:

- Poka-yoke (mistake-proofing operations);
- Statistical process control;
- Continuous improvement;
- Failure modes and effects analysis for both product and process;
- Line stop;
- Cell design (meaning, in this case, establishing work centers around natural work groups);
- Team roles, responsibilities, and rules;
- Graphic work instructions;
- Visual controls;
- Five "Ss." (The "five Ss" are *seiri, seiton, seiso, seiketsu,* and *shitsuke,* which roughly translate into English as "sifting," "sorting," "sweeping," "standardizing," and "sustaining." The first three terms refer to general housekeeping in the work cell. The last two terms refer to the self-discipline of workers to make the first three happen and the responsibility of management to see that they do.)

Most of these Lean tools have a direct application to project management, and TOC will help us identify which one to focus on for a particular project system. Dettmer also identifies the primary challenge of synthesizing Lean and TOC as related to two factors of Lean thinking: "Specifically, the overarching emphasis on cost reduction and maximizing local efficiency everywhere in the system needs to be rethought as Lean's focus on improving local optima." Our approach to CCPM builds on the strengths, while avoiding these obstacles.

2.3 Agile, or Light, Project Management

There has been quite a bit of attention paid to light, or agile, methods as a solution to the specific problems of projects involving information technology. The Wikipedia [7] notes, "Agile Methods evolved in the mid 1990s as part of the reaction against *high ceremony* methods, like Rational Unified Process (RUP), Prince and ISO 9000. The processes [that] originated from those methods were seen as bureaucratic, slow, demeaning, and contradicted the ways that software engineers actually work." Proponents sometimes characterize these approaches as Lean project management. The symptoms of the problems leading to these approaches were those described in Chapter 1: extensive cost and schedule overruns and failure to deliver error-free scope on most information-technology projects. Light methods for information-technology projects include such methods as

1. Rapid application development;
2. Joint application development;
3. Extreme programming;
4. SCRUM.

Detailing these methods is beyond the scope of this text.

I can't avoid a certain skepticism about some of the claims that lead to proposing the light or agile methods, such as "conventional project management does not work for information technology projects." I have not heard people objectively skilled in professional project management (e.g., certified PMPs) make these claims. For example, David Anderson [8, p. 55] notes,

> The traditional project management model focuses on locking the scope for a project and negotiating or varying the budget (including people and resources) and the delivery date. The PMI and ISO-9000 models for project management are based on this paradigm…[and] created the worst possible environment for managers…The result was heavyweight, traditional software methods…The existing PMI/ISO model for project management is obsolete. (p. 60)

Although Anderson goes on to present a masterful approach to deploying CCPM on information-technology projects, my own observations of information-technology organizations struggling with projects is that many do not understand and do not apply conventional project management well, if at all. Major shortcomings include poorly defined initial scope (e.g., lack of a WBS) and using ineffective change-management processes, or lacking them all together. Some of the

methods posed as alternatives to heavy project management, as characterized by the PMBOK™ Guide, are actually included in the PMBOK™ Guide processes, perhaps most notably the spiral-development rapid-prototyping approach. While the agile methods seem to provide effective approaches for leading small teams through small development efforts, I think of them more as supportive approaches to parts of larger projects than replacements for comprehensive approaches such as the PMBOK™ Guide and OMP3.

The following offers several perspectives on the issues of undefined requirements and agile approaches. First, standards such as the PMBOK™ Guide and OPM3 are not intended to be followed prescriptively in their entirety for all projects in all organizations. They represent menus to choose from and adapt to specific organizational needs. Although there is a natural human tendency to apply such standards prescriptively, and thereby to get bogged down in detail, that is not a fault in the standard: it is a problem with the application of the standard. I was fortunate to learn early in my career a simple way to adapt such standards to each project by specifying in the project plan the specific procedures that apply to each project. I learned to use checklists to quickly adapt the overall process to each project. Small, quick, inexpensive projects require very little formality and very simple planning and communication tools. Large, long-term, expensive projects involving multiple organizations require much more formal and extensive planning and control.

The second perspective deals with projects in which the requirements cannot be explicitly defined at the outset. Many information-technology projects fall into this category. Many other projects are not able to define all requirements at the outset, such as maintenance and repair projects or drug-development projects. In these cases, results from the early stages of the project change the tasks that must be accomplished later in the project. For such projects, the rolling-wave planning approach applies. You create specific project plans for that which you can plan with the information you have at hand (including assumptions), and you include in your plan the activities to update as new information comes available. Some organizations apply the rolling-wave approach with a long-term program plan, containing very little long-term detail and uncertain long-term projections to the end of the program. The program plan captures the major project-plan updates as a series of projects. These approaches exemplify the Lean maxim of eliminating the waste of planning that will not be used.

All projects require an effective change-control process to deal with the changes that will arise, including better definition of requirements. Change management is an essential part of agile project management. Key sections of the PMBOK™ Guide address change management. Information-technology project managers frequently complain about scope creep. I explain to them that my projects never experience scope creep and that I consider scope creep a self-inflicted wound by an inexperienced project manager who does not effectively apply change management. Most admit to not applying change management, frequently because they did not define the project assumptions or scope well enough initially or because they did not clarify the change-management procedures with project stakeholders at the outset. Once they start applying change management, they find that they are able to be more agile and achieve project success.

Finally, completing projects as soon as possible reduces overall waste, specifically the waste of changes that affect work that is already completed. CCPM supports maximizing the agility of project organizations to respond to changing needs, complementing all agile methods.

2.4 Six Sigma

Six Sigma, developed by Motorola but made famous by General Electric, adds to the approaches of TQM. The Malcolm Baldrige award represents the United States' symbol of highest achievement for business excellence. It grew out of a focus on TQM but today seeks to broaden its coverage. ISO 9000 is an international standard for quality performance, deployed by many companies. The Web site for the Malcolm Baldrige award [9] compares these approaches:

> Although all three are quality measurement systems, the Baldrige Criteria for Performance Excellence, ISO 9001:2000 Registration, and Six Sigma each offer a different emphasis in helping organizations improve performance and increase customer satisfaction.

Six Sigma

- concentrates on measuring product quality and improving process engineering.
- drives process improvement and cost savings.

ISO 9001:2000 Registration

- is a product/service conformity model for guaranteeing equity in the marketplace.
- concentrates on fixing quality system defects and product/service nonconformities.

Baldrige Criteria for Performance Excellence

- focus on performance excellence for the entire organization in an overall management framework.
- identify and track all-important organizational results: customer, product/service, financial, human resource, and organizational effectiveness.

The popular literature may lead you to believe that TQM was a management fad that failed to deliver on its promise and had outrun its applicability by the end of the century. A recent Six Sigma book asserts that it solves all of the problems that TQM experiences ([10], pp. 43–9). I consider TQM still to be quite successful, when applied appropriately, and Six Sigma to be part of the ongoing process of improving TQM.

The Baldrige criteria go beyond the Six Sigma literature in a number of areas, as noted above. At the February 1999 award ceremony in Washington D.C., the president of the United States noted that previous winners of the Malcolm Baldrige National Quality Award from 1988 to 1997 posted an impressive 460% return on investment, as compared to a 175% increase for the S&P 500 over the same period. Kevin Hendricks and Vinod Singhal [11] published results in April 1999 demonstrating that performance measures for TQM award-winning firms outstripping comparison control firms by two to one. For example, the TQM firms compared to non-TQM firms posted a 91% (versus 43%) increase in operating income, a 69% (versus 32%) increase in sales, and a 79% (versus 37%) increase in total assets. Although this performance dipped in 2002 due to the dominance by high-technology firms, it has risen again in 2004.

Six Sigma takes its name from a long-term goal of seeking to reduce defects, such that process output conforms with customer requirements within ±six sigma of mean process output, leading to a defect rate of less than 3.4 per million opportunities (the approach allows the mean output to float ±1.5 sigma). Sigma is the statistical measure of the process standard deviation. Six Sigma considers variation as the evil to focus on.

Six Sigma builds on Deming's plan→do→check→act (PDCA) cycle, defining the improved cycle as define→measure→analyze→improve→control (DMAIC). Six Sigma uses understanding of variation and statistical tools very heavily, key points we will focus on with CCPM. However, TOC takes the approach of using statistical knowledge to develop and deploy simple solutions, while Six Sigma tends toward rigorous application of the statistical tools. All of the Six Sigma approaches can complement applying CCPM, if you avoid the danger of suboptimizing on a single process versus focusing on exploiting the system constraint.

2.5 System of Profound Knowledge

Deming, the man whom most people consider the father of TQM, never defined it. Deming described his approach in seminars and books [12, 13] and, although he was a great advocate of operational definitions, chose never to offer one for TQM. Instead, he preferred to discuss the matter in terms of his fourteen points, or "Principles for the Transformation of Western Management." He supplemented these points with identified diseases and obstacles to achieving the transformation he preached.

In later life, Deming brought together the overall methods he believed in under the title of a "System of Profound Knowledge" [13], which he defined as a lens and map of the theory to understand and optimize organizations. He emphasized that profound knowledge is itself a system, having an aim and with all of its parts interconnected. He identified four segments for discussion but emphasized that they cannot be separated:

1. Appreciation for a system;
2. Knowledge about variation;
3. Theory of knowledge;
4. Psychology.

Figure 2.3 illustrates the interrelationship of the four elements. The following relates the elements to the project-management system.

2.5.1 Appreciation for a System

Every system must have a defined aim or goal. That is the purpose of the system and defines the boundary of the system. The systems itself is a network of interdependent components that work together to try to accomplish the aim of the system. Profit-making business systems have a goal to make money, now and in the future. That is why people invest in profit-making businesses. Nonprofit businesses (those intended to be that way, anyhow) have different goals, for example, creating health, for a health care institution, or creating family well-being, for some social institutions. Projects have the goal described above; to deliver the customer-specified, unique product or service on time and within budget. The client for that goal can relate the project result to the broader goal of the institution.

The project system consists of physical things, people, and nonphysical things, such as policies, knowledge, and relationships. All of these things are interconnected to varying degrees and may impact the performance of the system. Project planning and control is part of the project system. Task performance by the project team is part of the project system.

Things outside the system may impact it. Business systems are open systems, meaning that energy and physical things flow through them. Project systems are the same. These things flowing through, such as people, policies, and capital, can impact the system. For example, laws and regulations, which are outside both the business system and the project system, can have immense impact on the performance of the system.

Deming drew a sketch, similar to Figure 2.4, on a blackboard in Japan in 1950. He attributes the subsequent success of postwar Japan in large part the understanding conveyed by this figure. His description of the system starts with

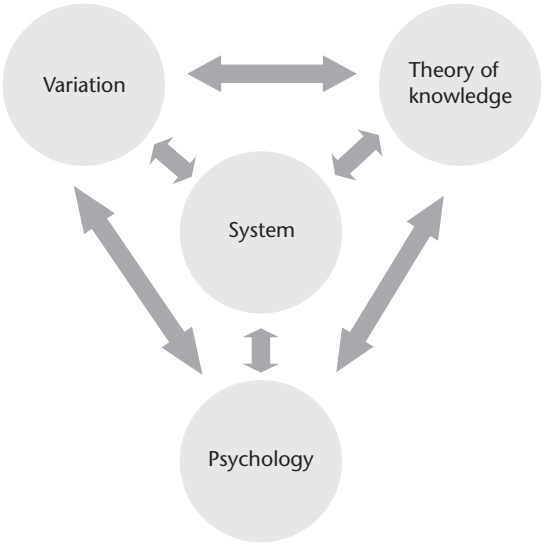


Figure 2.3 The four areas of profound knowledge interrelate.

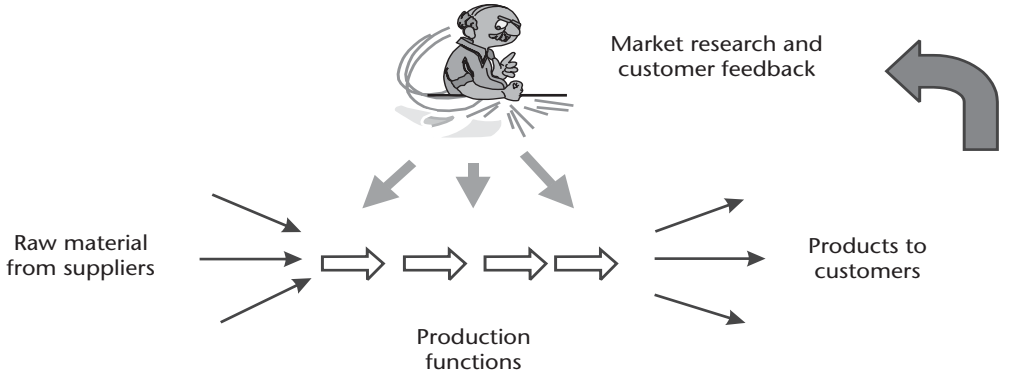


Figure 2.4 Deming's sketch of a business system emphasized interrelationships and feedback.

ideas about possible products or services. He considers these ideas predictions of what the customer might want or need. This prediction leads to the decision to design the product or service and to test it in preliminary trials before committing to full-scale production. Feedback from the customers is a key part of driving the system toward the future.

The project-management system operates in precisely the same way. Customers specify what they want from the project. The project team prepares a project plan to create the specified result. The plan brings together various functions within the company and purchased services and parts to produce the desired result. Just as a company may produce many products or deliver many services, the project-management system is capable of delivering many completed projects. Although the deliverables from specific projects are unique, the same project-management system serves to produce the results.

Deming understood system dynamics. He observed that operation of his flow diagram required the flow of material and information from any part of the system to match the input required by the next element in the system. He emphasized that the definition of the system must consider the impact on the future of the system. He made reference to the following material.

2.5.1.1 System Dynamics

Peter Senge [14] describes the essence of the discipline of systems thinking as a shift of the mind to

- Seeing interrelationships rather than linear cause effect chains;
- Seeing processes of change rather than snapshots.

He presents the Laws of the Fifth Discipline to summarize and understand how dynamic systems (including project systems) work. The following list gives the law and one instance of how it applies to the project-management system.

1. *Today's problems come from yesterday's "solutions."* (Management was unsatisfied with too long a schedule on the last project, so they cut the individual task estimates. This time, people added in a margin for management to cut out.)

2. *The harder you push, the harder the system pushes back.* (Management works to increase efficiency by assuring that all resources have multiple project tasks that they can work on. Working on multiple tasks means all tasks take longer since people can really only work on one task at a time. The others are doing nothing while one is worked on. Projects take longer and longer, and efficiency diminishes.)
3. *Behavior grows better before it grows worse.* (Management puts selected resources on overtime to accelerate the project. Results improve. The resources then get used to the extra income and slow down to not work themselves out of a job.)
4. *The easy way out usually leads back in.* (The "mythical man month" [15] explains this in detail. Management adds resources to recover schedule on a project that is slipping. Management must search for the people, hire them, create places for them to work, purchase them tools, and integrate them into the project team. This last step, in particular, requires the time of the most productive project resources. The project falls further behind.)
5. *The cure can be worse than the disease.* (The most common solution to improve project performance is to use more rigor and make more-detailed project plans. This often helps on a project just after a major project disaster due to regression to the mean; that is, it is unlikely that two projects in a row will have a bunch of bad breaks. So from then on, project plans are more complex and require more paperwork. Attention moves from completing the project tasks to completing the paperwork. Project durations and costs increase. Project changes increase, further increasing cost and time.)
6. *Faster is slower.* (The team passes on a piece of software that they really needed two more days of testing in order to meet their milestone date. The software causes problems in the integrated system test, which takes weeks to diagnose.)
7. *Cause and effect are not closely related in time and space.* (The space shuttle blows up on launch from the Kennedy Space Center in Florida. The cause is a seal design made and tested in Utah years before but not previously subjected to the specific environmental conditions. The Hubble space telescope is nearsighted (a billion dollar mistake) because crucial testing was skipped years before, on earth, to keep the schedule. This is the primary reason many of the studies on causes of project failure are incorrect. In complex dynamic systems such as projects, everything correlates in time. Cause and effect are impossible to determine without a model of the system. The TOC Current Reality Tree provides a tool for understanding cause effect relationships.)
8. *Small changes can produce big results, but the areas of highest leverage are often the least obvious.* (A major lever for systems containing people is the measurement and reward system. The impacts of measures and rewards are not always well thought out. For example, as Deming notes, monthly quotas lead to the "end of the month syndrome," where a lot of bad product is shipped. In projects, management emphasis that people "keep to their commitments" causes them to add time to their delivery estimates and withhold work that is completed early.)

9. *You can have your cake and eat it too, but not all at once.* (The critical-chain multiproject process completes more projects much faster; individual project duration decreases and the number of projects complete at any time increases. But, you must delay the start of projects to get the benefit.)
10. *Dividing an elephant in half does not produce two small elephants.* (Senge relays the tale of the blind men describing an elephant based on feeling its different parts: the trunk, the massive body, the leg, and the tail. Of course, their descriptions vary. Project-failure analysis often examines subprocesses, such as the work-plan process or the change-control process, to see what part of the system needs to be repaired. This approach fails to examine the underlying assumptions, for example, the assumption that task schedules are deterministic, which is implicit in printing out start and finish dates for thousands of tasks.)

2.5.1.2 Leverage

Dynamic systems lead logically to consideration of the possibility of using the system itself, as in jujitsu, to move the system in the direction you want it to go. Leverage defines small changes (inputs to the system) that cause large results (outputs from the system). The idea is like compound interest: a low interest rate can lead to a very large accumulation of capital if given enough time to work. People that knowingly work with complex systems focus on trying to find high-leverage interventions to cause desired outcomes.

Senge notes that there are no simple rules to find high-leverage changes to improve systems but that thinking about the underlying system structure, rather than focusing on events, makes finding these changes more likely. I contend that the major reason that there has not been a significant improvement in project management prior to CCPM is that all of the observers have been looking at the problem from the same flawed perspective (i.e., not looking at it as a system comprised of people, things, and information). They appear to have asked, how do I operate this system better. They should have asked, how do I improve this system?

Due to the effect of compounding, it is likely that any high-leverage interventions in systems will be in the feedback loops. Feedback loops affect the system based on the results that obtain. More results cause more feedback, so such loops are similar in impact to compounding interest. Powerful feedback loops for systems involving people always include the measurement and reward systems. Thus, the project-performance measurement system is one area where we may leverage improvements in the project system.

2.5.1.3 Unintended Consequences

The linkage and correlation between parts of a system means that changing any part of the system may influence other parts. As noted in the Laws of the Fifth Discipline, the change may or may not be in a desired direction, it may be large or small, and most likely it will not occur at the same time or in the same place as its cause. Many people, especially those prone to fiddling with social systems, talk in terms of "unintended consequences." Garrett Hardin [16] makes the point that from the ecological view of systems, there is no such thing as an unintended consequence. When you

change part of a system, other parts change. That's it! You can count on some of those changes being undesirable from one or more perspectives. Therefore, you must use caution when posing changes to a system such as the project-management system. Some of the changes posed to eliminate certain undesired results, or root causes, may have worse consequences elsewhere.

Several aspects of the project system illustrate this impact. For example, if we provide negative consequences for delivering a task result late, we will likely cause all subsequent estimates to include additional contingencies. We may also cause the quality of output to go down, influencing other tasks later in the project. We did not intend either of these effects, but they are predictable consequences of our action.

2.5.1.4 Destruction of a System

Destruction of a system by forces within the system was one of the key issues that Deming tried to bring home to management. He discussed how selfish competition verses cooperation between departments often causes such destruction. Senge [14] and Deming [13, 14] illustrate numerous examples of how government attempts to improve things often lead to destruction of the very system they hope to improve. For example, providing low-income housing often displaces job-producing industry while attracting more low-income people to the area, thus creating a larger problem than existed before the housing project started.

In project systems, these conflicts may arise between the client and the project team. They may arise between senior management in the company and the project team. They may arise between different parts of the project team. They may arise between the project team and supporting organizations within the company. A frequent example of the later is the nearly continual battle between procurement organizations and project organizations in large companies, especially those doing work for the federal government. Often the procurement organization's primary measures relate to compliance with a complex system of procurement regulations and policies, while the project team is only interested in having it fast and good. Sometimes the procurement organization's goal is to get it cheap, while the project organization wants it good. The project system design must ensure that the measures and rewards of individual parts of the organization cause these parts to work together to support the whole. Deming notes, "The obligation of any component [of a system] is to contribute its best to the system, not to maximize its own production, profit, sales, nor any other competitive measure."

2.5.2 Understanding Variation and Uncertainty

> I returned, and saw that under the sun, that the race is not to the swift, nor the battle to the strong, neither yet riches to men of understanding, nor yet favour to men of skill; but time and chance happeneth to them all.
>
> —Ecclesiastes 9:11

A project system attempts to predict and produce a certain result for a certain cost by a certain time. As the quote above illustrates, people know full well that the world is an uncertain place. Variation exists everywhere. Predictions are never

completely accurate. Indeed, the meaning of the word *accurate* is, in my opinion, not well understood when it comes to performing projects on time or within cost (see last paragraph of this section).

Understanding variation is essential to making any real system operate. Karl Popper [17], in an essay titled "Of Clouds and Clocks," describes a range of reality fundamental to understanding variation. He bids us to consider a horizontal line with a clock on the right representing the ultimate of a clockwork-like deterministic world. In this world, everything would eventually be completely predictable; it is only a matter of understanding completely the cause/effect relationships that determine the workings of this mechanical model. The ultimate manifestation of this model is the working of the planets of the solar system, whose motions are predictable with uncanny accuracy, using the equations defined by Isaac Newton.

The cloud, at the other extreme of Popper's continuum, represents complete chaos, not the deterministic chaos of current mathematics but the random chaos associated with the world of complete uncertainty. It represents the unpredictability of science at the quantum level and the unpredictability of nature on the human scale. Popper writes, "My clouds are intended to represent physical systems which, like gases, are highly irregular, disorderly, and more or less unpredictable." Everything falls between these two extremes.

Uncertainty refers to that which is indefinite, indeterminate, not certain to occur, problematical, not known beyond doubt, and/or not constant. All predications are uncertain. Fundamental physics tells us that all knowledge of reality is uncertain; the better we know the position of something, the less we know about how fast it is moving. Uncertainty is the true state of the world.

Most people use the words *variation* and *uncertainty* interchangeably. Dictionary definitions are not very helpful in making the distinction. For our purposes in this book, I will use *variation* as relating to getting different outputs from repeated application of the same process and *uncertainty* as relating to our knowledge about the result, as a measure of the predictability of the variation. For example, the results of any task in a project will vary if you do the same task over and over. Measuring this variation can produce an estimate of how much that task will vary in the future (e.g., the time or cost to produce the result will vary from project to project). You will use some method to estimate the task for a new project. That estimate will include the historical variation, plus introduce some other causes of uncertainty; for example, the people that do the task the next time may not be the some ones that did it before. With this definition, uncertainty is generally greater than historical variation.

Project managers can predict many things well enough to achieve the things they plan, such as building a house. Scientists also know that we can never accurately predict certain other things. For example, no matter how well we learn to model the weather or how well we measure conditions at a given time to run the model, the nature of physical laws limits our ability to predict specific phenomena, such as local weather behavior. Scientists now know (from chaos theory) that they will never be able to predict when and where the next tornado will touch down. On the other hand, they can predict seasonal trends reasonably well.

Starting in the seventeenth century, mathematicians and scientists have sought to move the ability to predict events the world further and further along Popper's continuum over into the cloudy region. At the same time, science kept expanding the

cloudy region to include more and more and of nature. It extended the smallest scale with quantum mechanics and showed cloudiness at the largest scale with increasing understanding of the universe. Cloudiness encompassed all intermediate scales with the discovery of chaos and the study of complex adaptive systems.

Common- and Special-Cause Variation Probability and statistics are science's weapons of choice to deal with cloudy systems. Walter Shewhart [18], a mentor to Deming, identified the need to operate systems in a state of statistical control in order to have a degree of predictability. He observed, "Every mathematical theorum involving this mathematically undefined concept [statistical control] can then be given the following predictive form: If you do so and so, then such and such will happen."

Following Shewhart, Deming emphasized the importance of distinguishing between common-cause variation and special-cause variation. He called failure to do so "a fatal error". It is necessary to distinguish them in order to get a system under statistical control. It is necessary to have a system under statistical control in order to predict its future performance. Common-cause variation is variation within the capability of a system to repeatably produce results. Special-cause variation extends beyond that range, usually due to causes outside the system. Management's function is to improve the system while avoiding two mistakes:

1. Treating common-cause variation as if it were special-cause variation;
2. Treating special-cause variation as if it were common-cause variation.

Deming called mistake 1 "tampering." Tampering causes unnecessary changes to a system that is operating in statistical control. Tampering always degrades the performance of a system. He described the case of a machine that had a feedback device attached to measure each part and adjust tool location based on that measure to try and improve the repeatability of each part. It made the variation in parts much larger because the measurements included the natural variation (capability) of the system to produce parts. The tool simply amplified this natural variation.

Tampering relates to the measurement and control of project performance and the decisions to take management actions based on those measurements. This phenomenon means that responding to common-cause variation as if it were special-cause variation will make the system performance worse. In other words, responding to small variances by making project changes degrades project performance.

All of the estimates in a project plan are uncertain. Performing each of the tasks within a project plan is a single trial of a system (the project-task performance system), and it is therefore impossible to predict the outcome with an accuracy better than the common-cause variation of the system. However, statistical techniques enable us to predict with known precision the likely results of numerous trials from a production system and to separate out the special causes of variation requiring corrective action. While knowledge of variation has been used to great profit in production operations, it has not (until now) been used to improve project performance. The PMBOK™ Guide and the supporting literature we have examined fail to differentiate between common-cause variation and special-cause variation. This is a major oversight in the current theory.

2.5.3 Psychology

Several properties of the human mind lead to individual behavior that seems to resist change. B. F. Skinner [19] describes one of the more powerful mechanisms. Skinner asserts (with extensive scientific data) that much human behavior comes from "operant conditioning." Put simply, this means you continue to do what brings positive reinforcement, and learn to stop doing things that do not lead to positive reinforcement, or that bring negative reinforcement. Positive reinforcement is something you like. Negative reinforcement is something you don't like. Positive and negative reinforcers vary from individual to individual. Skinner notes, "A reinforcing connection need not be obvious to the individual reinforced."

Figure 2.5 illustrates the author's rendition of a control-system view of Skinner's model. It starts with a need, which is influenced by the person's present state, including deprivation or satiation relative to the goal. Comparing this need to the person's understanding of his or her current situation (perceived reality) yields a gap that, if large enough, motivates a person to action. Action seeks to change reality to close the gap. The sensor, which may be one of the five senses or a more removed method of gaining data, feeds back information about the effect that this action has on reality. If the change is positive (reducing the gap, or otherwise supplying a "reward"), it strengthens the chances that the person will repeat the behavior. This is what Skinner calls operant conditioning.

This operant conditioning must somehow be stored within the brain. Since it defines a (perhaps rudimentary) model of the world (i.e., if I do this, then I get that), you can consider it a belief about how the world works. Such beliefs may be conscious or unconscious. Research demonstrates that these beliefs have other impacts on the model. Figure 2.5 illustrates that beliefs impact what we pay attention to, how we interpret what we sense (perception), what our motivations (needs) are, and the decisions we make about to act in the world so as to increase our

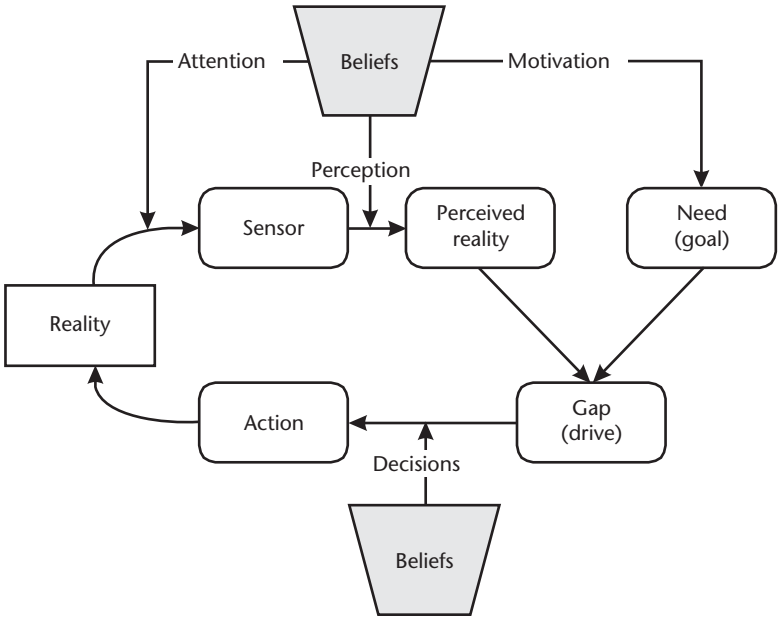


Figure 2.5 Control-system view of human actions (behavior).

rewards and decrease our negative reinforcers. This influence is mostly unconscious. In other words, you see it because you believe it.

Rewards While operant conditioning works well for rats and pigeons, you must use extreme care applying the model to human beings. Much of the damage done in organizations follows directly from applying oversimplified models of operant conditioning to humans. The field of performance measurement and concepts such as pay for performance are just some of the worst examples of ineffective practices derived from oversimplified application of reward–punishment concepts, even though Skinner identified, described in depth, and proved by experiment that punishment does not work.

Worse yet, research with humans demonstrates, conclusively and repeatedly, that rewards only work to motivate people to get the reward. Usually there are more unintended negative consequences from reward systems than positive benefits. Alfie Kohn [20] describes the reasons for this, noting that reward and punishment are really two aspects of the same thing: attempts at external control. He explains five reasons why rewards fail:

1. Rewards punish.
2. Rewards rupture relationships.
3. Rewards ignore reasons (for the problem that elicited the need for a reward).
4. Rewards discourage risk taking.
5. Rewards cause people to lose interest in the task itself and therefore to lose intrinsic motivation.

This is not news, but much of modern management does not get it. Frederick Herzberg [21] noted,

> Managers do not motivate employees by giving them higher wages, more benefits, or new status symbols. Rather, employees are motivated by their own inherent need to succeed at a challenging task. the manager's job, then, is not to motivate people to get them to achieve; instead, the manager should provide opportunities for people to achieve so they will become motivated.

The requirements for CCPM must include designing the system to provide these opportunities. A significant barrier in the deterministic critical-path approach is that workers win or lose depending on whether they complete their tasks on time. Yet, all involved know full well that the task duration estimates in the schedule have significant uncertainty. As Deming demonstrates with his bead experiment [7], random fluctuations determine employee success or failure. This system clearly does not meet the design requirement.

Additional Psychological Considerations One modern view focuses on how our minds operate as pattern-recognition devices. You have a wonderful ability to infer the automobile in the picture by looking at only a small fragment of the picture. You can often name a tune in three notes. It's remarkable, when you think about it.

Beliefs act to focus our attention, and they adjust our perception of reality by acting as a kind of information filter. Two people witnessing the same events may have dramatically different views of what really happened. I was fascinated while

listing to congressmen from both parties arguing about the impeachment of President Clinton. Participants from both sides made very logical and emotional arguments for their positions. No one argued that they held their position because of the political party they were aligned with. Yet, when the vote came in, only five representatives of 417 crossed the party line in their vote. While I am certain that a small minority literally chose to vote with the party, the speakers convinced me that they really believed the logical arguments that they made for their side. Since the argument was framed as an either-or choice, one would expect that arguments based on factual analysis should have aligned people regardless of political party. My perceptive filter saw this as an outstanding example of how people interpret the facts (i.e., perceive) in ways that align reality with their beliefs. The impassioned logical arguments of both sides had no impact whatsoever on the other side because they did not change the basic underlying beliefs. The participants in the debate were each locked into their paradigm.

People operating in any environment tune their behavior to the environment. Put another way, feedback through operant conditioning causes them to behave in ways that maximize positive reinforcement and minimize negative reinforcement in the current environment. Changes in the system threaten this equilibrium. Furthermore, Skinner demonstrates that extinguishing behavior established by operant conditioning can take a long time. The organism will continue to perform the old behavior, which is no longer reinforced, sometimes for thousands of tries!

Other aspects of psychology, or how our minds work, are also important to understand the system you are attempting to change. One of these is the availability bias. Psychological experiments repeatedly demonstrate that people are relatively poor judges of probability. Instead, people focus on the information they heard or saw last or that impressed them the most when offering judgments about probability. Thus, for example, you will often hear statements such as, "All scientists (programmers, engineers, etc.) tend to underestimate how long it will take to do a task." When pressed for data, people admit to having little. Data analyses often prove otherwise. My analysis of data from several organizations illustrates that people report most project tasks as complete on the due date. (A miraculous occurrence, by the way, proving the existence of date-driven behavior.) I also continue to hear project-task status reported as on schedule, meaning that a date or duration of some probability has been converted into a deadline. We will dig into the implications of this a little further on. Many studies also show that people also tend to be overconfident in their ability to estimate ranges of data or probabilities.

The PMBOK™ Guide does not deal directly with psychology as a knowledge area. Despite this, many project-management texts deal with the human side of project management. The project system must integrate with the human subsystem. This integration happens through the psychology of individuals and groups. Since the present system was not designed with this connection in the forefront, you may expect to find some problems in this area. Chapter 3 demonstrates that the core conflict leading to most of the observed undesired effects (UDEs) with the current project system stem from a mismatch between individual psychology and the project system goal.

2.5.4 Theory of Knowledge

Popper [17], in an essay titled "Conjectural Knowledge," states, "From a rational point of view, we should not rely on any theory, for no theory has been shown true, nor can be shown to be true." This point, agreed upon by most philosophers and scientists, does not reflect the understanding of the common man, who is prone to accept a single instance that conforms to a theory as evidence that the theory is right. Popper goes on to state,

> In other words, there is no "absolute reliance"; but since we *have* to choose, it will be "rational" to choose the best tested theory. This will be "rational" in the most obvious sense of the word known to me: the best tested theory is the one which, in the light of our *critical discussion*, appears to be the best so far, and I do not know of anything more "rational" than a well-conducted critical discussion.

He also suggests that an objective criterion to prefer a new theory "is that the new theory, although it has to explain what the old theory explained, corrects the old theory, so that it actually contradicts the old theory: it contains the old theory, but only as an approximation."

Figure 2.6 illustrates the scientific method. The method operates based on effect→cause→effect. Scientists start by defining a problem: hypothesizing the cause for an observed effect. All new theories have some confirming evidence; that is why scientists propose them. The prediction of a previously unseen effect that differentiates the new theory from the old tests the theory. Existence of the predicted effect provides evidence to prefer the new theory to the old. Lack of the effect fails to provide evidence to prefer the new theory. A theory is usable until disproved. A successful experiment does not mean that it is correct (true), and it does not mean that it will work into the future. A successful experiment just means that it worked over the domain so far experienced.

Newton's laws of motion and gravitation are commonly used as examples of the scientific method. Before Newton, mankind gathered much data on the positions of the sun and planets. They developed correlations to accurately predict the motion of the heavens. There was a fundamental flaw, of course, in that they had the earth at the center of the solar system. Nevertheless, the predictions worked.

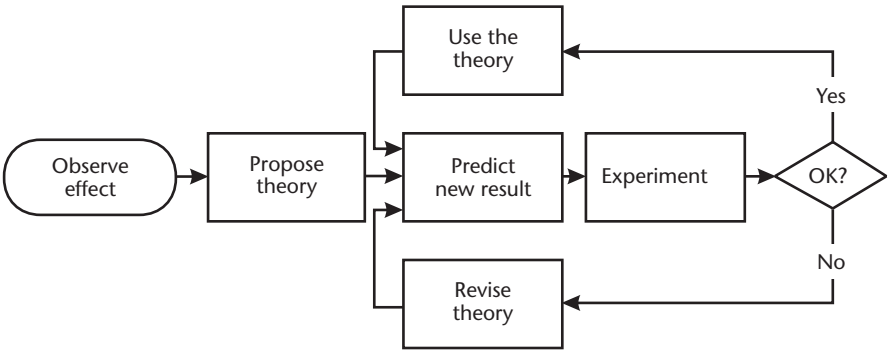


Figure 2.6 The scientific method checks the validity of a theory by experiment. No theory is ever proven. It is accepted as good enough to use until rejected by a single experiment or replaced by a theory that better predicts reality.

Newton's laws worked better than his predecessors did because they extended beyond observation. Newton's laws allowed prediction beyond the realm of the observed and have enabled us to put men on the moon and send spacecraft to Jupiter. This would have been impossible using the correlation of planetary movement.

Then, along came Einstein. His equations proved that Newton's equations are wrong. (Newton knew this also; he proposed them as "good enough.") Einstein's equations reduce to Newton's equations where speeds are modest compared to the speed of light and where gravity is not too large. Inclusion of the previous theory as a special case fits Popper's model of a better theory. Einstein spent his later life trying to prove his own theory wrong by developing a unified theory. So far, no theory better than Einstein's theory has been found. Therefore, scientists continue to use Einstein's theory. This is a theory of knowledge at work.

Understanding the theory of knowledge enables you to better test the CCPM theory compared to the critical-path theory or another theory of project management you are currently using. You now know you can never prove a theory true, but you have working tools (test and critical discussion) to choose between competing theories. The theory of knowledge will also help you make decisions necessary to plan a specific project and to operate the project system you choose.

2.6 Theory of Constraints

The basic TOC is a commonsense way to understand a system. TOC says that any system must have a constraint that limits its output. You can prove it with critical discussion. If there were no constraint, system output would either rise indefinitely or would fall to zero. Therefore, a constraint limits any system with a nonzero output. Figure 2.7 shows that limiting the flow through any of the arrows can limit the total output of the system. That arrow would be the system constraint. People identify the constraint in physical systems as a bottleneck, a constriction limiting flow through the system.

The purpose of using the TOC is to improve a business system. In *What Is This thing Called Theory of Constraints* [22], Goldratt states, "before we can deal with the improvement of any section of a system, we must first define the system's global goal; and the measurements that will enable us to judge the impact of any subsystem and any local decision, on this global goal."

Deming noted in *The New Economics* [13], "We learned that optimization is a process of orchestrating the efforts of all components toward achievement of the stated aim."

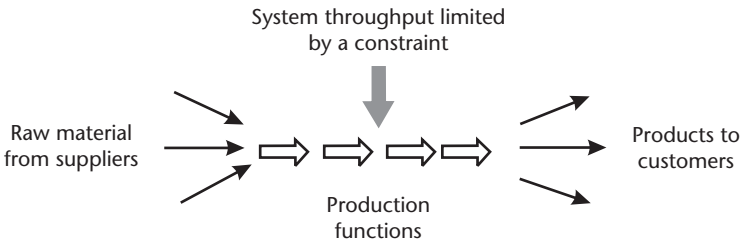


Figure 2.7 TOC limits the output of a system by a constraint.

A physical chain provides the most commonly used prop to describe TOC (Figure 2.8). The goal of a chain is to provide strength in tension. Everyone accepts that the weakest link determines the strength of a chain. Anyone can see that improving the strength of links other than the weakest link has no impact on the strength of the chain.

The next step in understanding TOC is not so evident. TOC makes a leap to throughput chains, and poses the theory that for any chain, throughput (at any time) is limited by at most one constraint. Perhaps this is easier to see in the project world, where a project plan can have only one longest path. The only case where this would not be true is if two or more paths are exactly the same length. As soon as you start to perform the project, it is likely that one path will become the real constraint. The constraint (longest path) will seem to shift due to fluctuations in project activity performance. But at any time, only one controls the actual time to complete the project.

Applying the scientific method to this basic understanding of the TOC leads to many principles. Dettmer poses the following list in his book *Eliyahu M. Goldratt's The Theory of Constraints, A Systems Approach to Continuous Improvement* [23]:

1. System thinking is preferable to analytical thinking in managing change and solving problems.
2. An optimal system solution deteriorates after time as the system's environment changes. A process of ongoing improvement is required to update and maintain the effectiveness of a solution.
3. If a system is performing as well as it can, not more than one of its component parts will be. If all parts are performing as well as they can, the system as a whole will not be. The system optimum is not the sum of the local optima.
4. Systems are analogous to chains. Each system has a "weakest link" (constraint) that ultimately limits the success of the entire system.

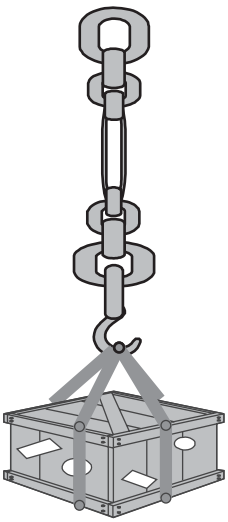


Figure 2.8 A physical chain illustrates TOC in action: the weakest link constrains the strength of the chain.

5. Strengthening any link in the chain other than the weakest one does nothing to improve the strength of the whole chain.
6. Knowing what to change requires a thorough understanding of the system's current reality, its goal, and the magnitude and direction of the difference between the two.
7. Most of the UDEs within a system are caused by a few core problems.
8. Core problems are almost never superficially apparent. They manifest themselves through a number of UDEs linked by a network of effect→cause→effect.
9. Elimination of individual UDEs (undesired effects) gives a false sense of security while ignoring the underlying core problem. Solutions that do this are likely to be short-lived. Solution of a core problem simultaneously eliminates all of the resulting UDEs.
10. Core problems are usually perpetuated by a hidden or underlying conflict. Solution of core problems requires challenging the assumptions underlying the conflict and invalidating at least one.
11. System constraints can be either physical or policy-based. Physical constraints are relatively easy to identify and simple to eliminate. Policy-based constraints are usually more difficult to identify and eliminate, but they normally result in a larger degree of system improvement than the elimination of a physical constraint.
12. Inertia is the worst enemy of a process of ongoing improvement. Solutions tend to assume a mass of their own, which resists further change.
13. Ideas are not solutions.

TOC also undergoes continuous improvement. When Goldratt introduced the Thinking Process, the method to locate what to change in a system relied on discovering the *core problem*, as illustrated by the above list. The removal of a core problem will begin to cause the system to change UDEs into DEs (desired effects). In other fields, it is called the root cause. Goldratt later shifted to define a *core conflict* instead of a core problem. This is a significant step in the theory. It claims that most of the UDEs in a system flow from an unresolved, or at least unsatisfactorily resolved, conflict or dilemma. Substituting core conflict for core problem into the above list (except for item 10) makes it reflect present understanding. Item 10 in the list is the earlier statement of the previous understanding.

The idea of a core conflict underlying many system UDEs must rest on the thought that people would change the system to eliminate UDEs if they knew how and were able to make the changes. If UDEs persist in a system, then something is preventing the system designers or operators from changing the system to eliminate the UDE. The core conflict idea helps to identify that something.

TOC applies a tool called current Reality Tree (CRT) to logically link the core conflict to the UDEs.

For reasons I will detail later, I recommend getting as wide a range of perspectives as possible both when building and analyzing CRT. The review process within the organization will work to eliminate the mismatches between actual behavior and beliefs, so it will take concerted effort to communicate and correct the inevitable gaps.

2.6.1 The Throughput World

Goldratt found that most of the time, system constraints trace back to a flawed policy rather than to a physical constraint. In *The Goal* [24], he demonstrated that these policy constraints derived from a flawed system of accounting. Accounting systems in use today trace back to the turn of the century and have changed little since (twice in the history of project-management systems), When they were developed, they were based on assumptions (no longer listed) about the design of business enterprises.

Goldratt defined the old accounting system as the Cost World because it operates on the assumption that product cost is the primary way to understand value and make business decisions. This requires the allocation of many expenses to products through elaborate product cost schemes, such as activity based costing. These schemes are full of assumptions and often lead to erroneous understanding and decisions.

Goldratt defined a new way of accounting, called the Throughput World. It rests on three definitions:

1. Throughput (T): all of the money you make from selling your product (revenue minus raw material costs);
2. Inventory (I): all of the money you have tied up in fixed assets to enable you to make the throughput (the primary difference here is that fixed assets and work in progress inventory are treated the same);
3. Operating expense (OE): all of the money you spend to produce the throughput.

Major accounting authorities around the world have endorsed this method, but it has been slow to penetrate the main stream.

The Cost World was not so bad when it was developed around the turn of the century. At that time, big business (which designed it) consisted primarily of plants with very large capital investments (e.g., resource industries, steel, railroads, and a little later, automobile manufacturing), representing fixed cost. At that time, things were tough for labor; labor was a variable cost. Labor was mostly applied to very unskilled jobs and, therefore, plentiful and easy to replace. Therefore, it was easy to vary the workforce with demand.

Today, the skilled workforce is much less variable, and the traditional fixed costs are much less fixed. The concept of allocating costs to labor or products always requires many arbitrary assumptions. These assumptions, often long forgotten, influence the business decisions made using the cost-accounting practices.

The Throughput World corrects these errors and focuses all decisions on the goal of the company (i.e., to make money now and in the future). All decisions and measures relate to the global goal. These often lead to different decisions than those dictated by the Cost World.

For example, in the Cost World, managers measure the operating efficiencies of local workstations. Financial people count inventory as a company asset. If they do not need workers to produce product for customer need, then they produce product for inventory, increasing efficiency to make themselves and their local plant look good. Unfortunately, the plant does not make money on inventory. Inventory costs

money to buy (raw materials) and to store, so it hurts cash flow and reduces disposable cash at the plant. This accounting system counts inventory as a good thing (an asset), but it's bad for business. Then, when you get around to selling the inventory (which is good), it reduces your assets (which can look bad). If you can explain why all this makes sense, please write to me.

On the other hand, what do people normally consider their biggest competitive edge in knowledge industries? People. What are people in the accounting system? Expenses. They look bad. They are the first things you want to get rid of if business looks bad; keep the assets, drop the expenses. Dump your ability to make money now and in the future; keep your hardware, which costs you money.

An effective way to evaluate the dilemma facing managers is to apply one of the Thinking Process tools invented by Goldratt, the Evaporating Cloud. Figure 2.9 illustrates the Throughput World/Cost World Evaporating Cloud. Block A represents a common objective all managers share. Blocks B and C are requirements to achieve the objective. You read the cloud, "In order to manage properly, managers must control cost." You read the lower branch, "In order to manage properly, we must protect throughput." So far so good.

Focus on throughput requires understanding and controlling the whole system to optimize throughput. The most important effect of Throughput World thinking is that it requires focus on throughput as the much-preferred path to system improvement. Looking at how *T*, *I*, and *OE* impact net profit and return on investment leads to the immediate conclusion that *T* is the most important variable. Improvements in *T* are unbounded, while improvements in *OE* and *I* are limited.

Cost World thinking leads to a piecemeal view of each part of the production system. Costs add algebraically. The Cost World leads to a focus on *OE*. You can reduce *OE* in any part of the system, and the *OE* reductions add up. This thinking leads to entity *D*, with the logic, "In order to control cost, managers have pressure to manage according to the Cost World." Why hasn't everyone adopted throughput accounting and thinking? Inertia. Chapter 3 explains the importance of thinking and operating in the Throughput World relative to projects.

2.6.2 The Production Solution

Goldratt's first career was as a developer of computer software for factory management. He built a very successful business, and his clients were quite satisfied with the

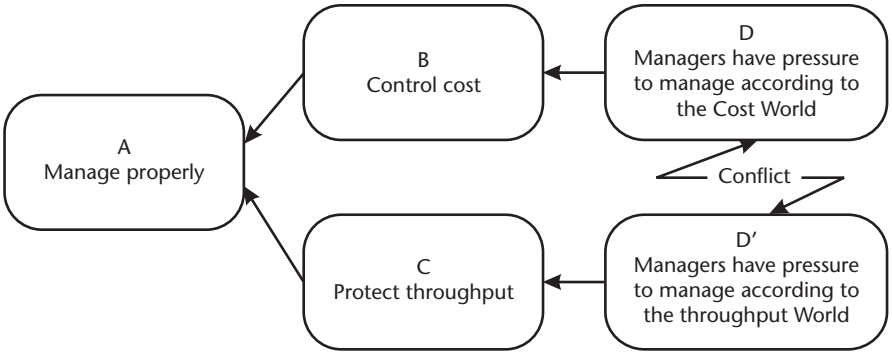


Figure 2.9 The Throughput World/Cost World Evaporating Cloud exposes the manager's dilemma.

software; it gave them much-more-detailed information about where things were in their factories. He noticed after a while, however, that they were not making any more money using his software. He thought about this and realized that he had to derive the basic principle from a focus on the goal of a for-profit company (i.e., to make money now and in the future). The goal corresponds to what Deming means by the aim of a system.

Goldratt's books, most notably his initial international best seller *The Goal*, demonstrate how he invented and used the TOC to develop the elegant Drum–Buffer–Rope method for controlling production. The Drum–Buffer–Rope method is elegant because it is much simpler than the earlier methods of production management that attempted to control the production system through detailed complexity. The Drum–Buffer–Rope system focuses on the dynamics of the production system.

The *drum* is the processing capability of the constraint. It determines the overall throughput of the production process. Recall that throughput is the difference between sales revenue and raw material costs. To exploit (make maximum use of) the constraint in terms of throughput, you have to release the correct work into the system at the proper time in order not to ever starve the constraint and also not to overload it. Overloading the constraint (i.e., producing more than it can process) creates excess in process inventory (i.e., piles of incomplete work in front of the constraint). The *rope* transmits information from the drum to the release of work in order never to starve the constraint and to limit the build up of inventory.

Buffers are deliberate placement of in-process inventory to account for statistical fluctuations in the process system. Machines break, go out of alignment, or sometimes need unplanned maintenance. People do not always show up on time and do not work to a constant rate. The buffers account for these fluctuations.

Figure 2.10 illustrates a production system. Compare it to Figure 2.4 and note that this represents the inner workings of the overall business system depicted by

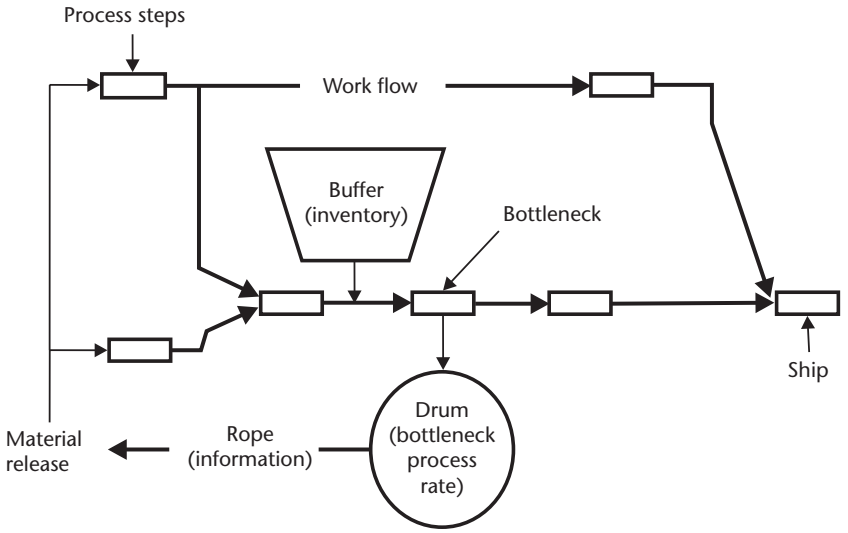


Figure 2.10 Drum–Buffer–Rope is the solution to operating a production facility using TOC. This solution operates to the global optimum (the system goal) and accounts for the combination of statistical fluctuations and dependent events.

Deming. Production is a subsystem of the overall business system, just as the circulatory system is subsystem of the human body.

Although Goldratt uses as a background in *The Goal* a factory that produces hardware products, the general nature of figures 2.4 works for any kind of system. The output is anything an organization does that it sends outside. Output includes scientific-research results, services of any kind, meetings, travel arrangements, reports, legal aid, software products, or any other output of any profit or non-profit organization. The systems mentioned also include government. Nonprofit and government systems obviously have a different goal (aim) than for-profit business.

Figures 2.4 and 2.10 are static pictures of a production system. The system stays fixed. Inputs flow through the system, converting to outputs. The flow through the system is not uniform. Each step in the processes has some amount of variation, often referred to as statistical fluctuation. Since workstations downstream of other workstations need the parts from the upstream workstations, they depend on the upstream workstations. This combination of dependent events and statistical fluctuations is an important issue in managing the overall system, especially at the constraint.

A system designed with capacity for steps upstream of the constraint equal to the capacity of the constraint cannot produce at the capacity of the constraint. The reason is that upstream fluctuations add up, leading to periodic starving of the constraint. The constraint can never make up this lost production because it is the constraint of the system. Therefore, all upstream workstations must have excess capacity in an optimum system.

Likewise, all workstations downstream of the constraint must have capacity that exceeds the capacity of the constraint. Otherwise, they can never make up any downside fluctuations in their performance relative to the performance of the constraint. Most of the time, they operate at the capacity of the constraint (the drum for the system), but the excess capacity allows them to catch up when necessary. This means all nonbottleneck machines in a production facility should spend some of their time not working.

This reasoning extends to the conclusion that *a system operating with each step at optimum efficiency cannot be an efficient system.* Most people intuitively believe that operating each part of a system at maximum efficiency causes the system to operate at maximum efficiency. You can see that an optimum system has to feed the bottleneck at its capacity and process the downstream parts at the bottleneck's average processing rate. This means that, on average, every nonbottleneck process must operate at lower efficiency than the bottleneck in order to have reserve capacity to make up for fluctuations.

This understanding is a major reason that the TOC is able to make such immediate impact, once people understand it. Managers design and operate most current systems without a critical understanding of the TOC. They work to cut costs everywhere, including the capacity of the constraint. They work to improve efficiency everywhere, including workstations upstream of the constraint that may cause the constraint to work on things that do not translate into short-term throughput. Once they understand the theory, identify the constraint, and improve its throughput, the system throughput increases immediately.

The computer systems that Goldratt was selling before he invented the TOC, as well as all other factory control systems, failed to account for the impact of the system constraint combined with statistical fluctuations and workstation dependency. Since the actual fluctuations are statistical, they are unpredictable. You can only predict the general behavior over a period of time for many items that flow through the system. Therefore, the schedules produced by the computer systems were out of date and incorrect as soon as they were produced. No wonder the scheduled did not cause the system to make more money. No wonder adding more detail to project plans did not make projects more successful.

In *Critical Chain*, Goldratt extended the concept of Drum–Buffer–Rope to project planning and performance. It is not a direct extension because project work on activities moves through time, while in a production facility, the parts move through fixed workstations. The same constraint phenomena apply to projects. The combination of statistical fluctuations and dependent events exists in a project. Current computer planning and control methods do not consider these fluctuations. Therefore, many of the same phenomena take place in projects that took place in production before Drum–Buffer–Rope (i.e., late delivery, longer and longer delivery times, resources not available when needed, and so on). More detailed planning or more sophisticated computer programs cannot correct these problems because of the structure of the project reality. You do not reduce uncertainty by cutting up tasks (remember the Fifth Discipline law about elephants). More-detailed plans increase static complexity but do not help deal with dynamic variation due to uncertain estimates.

For a project, the *critical chain is the constraint*. It is the focus for management of the system. The buffers are time buffers instead of material buffers. (Actually, in production the physical material buffers relate to time also. A pile of a certain size provides a certain period of protection for the machine that works on the pile.) Buffer management for projects is similar to the production counterpart. Counterparts to the rope are

- Release of projects to the system based on the ability of the constraint resource to process the project tasks;
- Release of activities for work based on the input from buffer reporting;
- The decisions made in buffer management on when to change the process (buffer recovery).

Many people have found it difficult to apply TOC understanding to their work. They can see from *The Goal* how to apply it to a physical production system but cannot see it in their system, which may be a service business, research-and-development (R&D) firm, a nonprofit organization, or a government agency. A middle manager of a current client recently stated, "Work in [our business domain] is way too complex. CCPM is doomed to fail given the inherent complexity of our work." (I have other clients successfully applying CCPM in organizations 40 times as large, with projects 100 times as complex.) There is no basis for a distinction based on the type of business: the theory applies to any business system and, so far, to every type of project people have tried it on—an extremely wide range of project types. You should expect more dramatic improvements for projects with greater

uncertainty, but all organizations have projects with a range of uncertainty. *It's Not Luck* [25] shows how TOC tools apply to marketing, personal career planning, and personal issues at home.

Experience demonstrates that even in production systems, the constraint usually turns out to be a policy, not the physical bottleneck. *The Goal* [24] demonstrates this relative to financial and sales policies.

Consider a service business that answers telephone calls from customers. A common measure for such services is the number of calls per hour handled by each person. The goal of the system does not relate to the number of calls but to some effect from answering the calls (e.g., satisfied customers or orders taken). Calls have statistical fluctuations in their length, and they arrive at random times. Let us suppose you are a customer and want to order many things. Should the operator keep you on the line and get marked down for taking fewer calls per hour? How long will you wait for an operator to answer before you call a competitor?

As the manager of this service, how do you decide when to get more operators? If you have excess operators (so that the longer calls and variations in when calls arrive can be handled), your efficiency goes down, even though the throughput for the company may go up far more than the added operating expense. What is the constraint to this system?

Consider another case representative of many internal functions in a company, the human resource function. What is your department goal, and how does it relate to the company goal? How do you measure output to ensure that you are contributing to the company goal? Do you know where the company constraint is and how human resources might influence it? Goldratt defines several necessary conditions for achieving the goal of a company. One of these is to "Satisfy and motivate employees now and in the future." This condition directly affects the throughput of the company. Human resources clearly affect this necessary condition. Human resources also impact operating expenses in several ways, including their own contribution (cost) and their effect on company salaries and benefits through salary and benefit policies and union agreements.

2.6.3 Five Focusing Steps

Having realized the goal of the system and the fact of a constraint, Goldratt invented the five focusing steps as a process to get the most out of a system in terms of the system goal. Figure 2.11 summarizes these steps.

2.6.3.1 Identify the System's Constraints

In order to improve the system in terms of the goal, you have to identify what is holding it back. You have to decide "what to change." The system's constraint is like the weakest link in a chain; no matter what we do to improve other links in the chain; the chain does not become stronger until you improve the strength of the weakest link. It is evident that you have to find the weakest link before you can improve it.

In a project-management system, the weakest link can be anywhere: in the project-management process, in company management policies, in any of the supply chains, in work procedures, in the measurement system, or in communication. Since

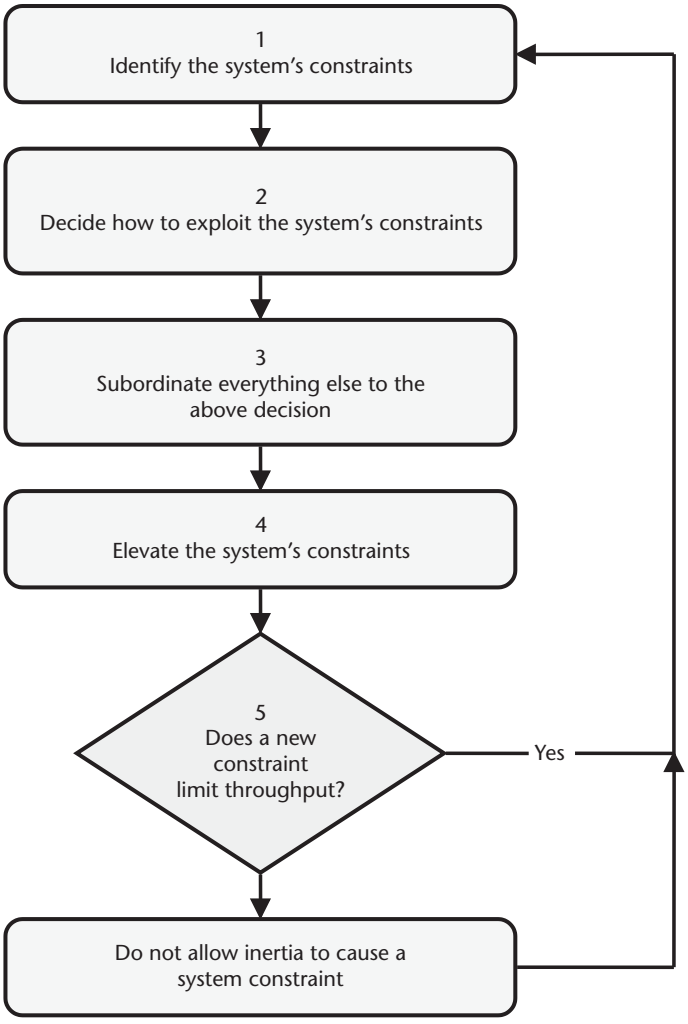


Figure 2.11 The five focusing steps represent the TOC approach to ongoing improvement.

a project does not have physical form until it is well under way, the constraint is often not evident. Systems theory describes why and how symptoms may occur a long time after the actions that caused them. (See the Laws of the Fifth Discipline.) You also know that the symptoms may appear somewhere other than the cause, through chains of effect→cause→effect. Therefore, study of why projects have gone wrong may not identify the actual cause of the symptoms.

TOC identifies the constraint of a nonproduction system as a core conflict. Like any constraint, the core conflict is the primary reason that the system is not performing better. It is the root cause of one or more UDEs in the system. In order to eliminate these UDEs, you have to first identify the core conflict.

2.6.3.2 Decide How to Exploit the System's Constraints

Exploiting the system constraint requires getting the most out of the weakest link of the chain. There are usually a variety of ways to do this. For example, in a production facility one way to improve throughput of the production system is to change

the way the system puts things through the bottleneck (constraint). It must assure that policies maximize using the constraint in terms of the goal. For example, ensuring the quality of parts entering the bottleneck prevents the bottleneck from wasting time on defective parts. The schedule ensures that products with the closest delivery date complete first.

For a nonproduction system, you have to decide how to eliminate the core conflict and assure that you change the necessary parts of the system so that the natural effect→cause→effect that results from these changes will achieve the DEs.

In this step, you are deciding "what to change to."

2.6.3.3 Subordinate Everything Else to the Above Decision

This is the key to focusing your effort. While subordinating, you may find many assumptions that seem to inhibit doing the right thing. For example, in *The Goal,* Alex Rogo discovers measurement constraints (efficiencies) that would prevent him from doing the right things if he paid attention to them. The accounting system valued finished-goods inventory as an asset, and it made his financial reports look good to build inventory. In fact, making and storing inventory costs money and can plug up the system's constraint, delaying work that would otherwise go directly to a customer and create income. Likewise, measuring workstations by efficiencies causes people to build parts for products that are not going to sell immediately, causing cash outlay for parts, and possibly plugging the system constraint, again impacting products that customers want and that would lead to immediate income.

Since project management has been in existence for over 40 years with little change, isn't it likely that there are some assumptions, policies, or artificial constraints that do not work well any more? Is it possible that some of the measures used to manage a project actually make it less likely to meet the goal?

This step is the first part of deciding "how to cause the change."

2.6.3.4 Elevate the System's Constraints

This is the implementing part of "how to cause the change." It is often the most difficult part, not difficult because of the physical work but because of the changes it demands in how people look at things. After all, they have been doing things the other way for a long time without questioning their assumptions. People naturally defend what they have always done.

Sometimes, this defensiveness prevents us from even conceiving of different ways of doing things. It always makes it difficult to implement something new.

2.6.3.5 If in the Previous Step a Constraint Has Been Broken, Go Back to Step 1

As you continue to elevate the current constraint, you always eventually unearth another constraint. It may be lurking a few capacity percentage points above the current constraint, or you may be able to improve the system by many tens of percentage points before you uncover the next real constraint. This is not a problem, it just provides a natural strategy to follow in improving a system: always focus on the current constraint. This is the optimum continuous improvement strategy.

A strong caution follows these steps is not to let management's inertia become the system's constraint.

2.6.4 The Thinking Process

Figure 2.12 illustrates the overall Thinking Process flow and identifies the primary tools. Goldratt designed the Thinking Process to answer three questions:

1. What to change;
2. What to change to;
3. How to cause the change.

The process steps link so that the output of each step provides the input for the next step.

Goldratt developed the tools necessary to apply the Thinking Process. In addition to their use in the Thinking Process, the tools (other than the Current Reality and Future Reality trees) have stand-alone application. The text below describes the tools, but it will not use most of them until near the end in order to keep the text accessible to readers who may not be interested in learning more about TOC but would like to improve their projects. The text does make extensive use of the Evaporating Cloud, the most elegant stand-alone TOC tool. The final chapter demonstrates application of the Thinking Process to create CCPM. Dettmer [23] provides an effective description and set of procedures to apply the Thinking Process.

Many people find the list of TOC Thinking Process tools and associated acronyms intimidating at first. Most people require two to three weeks of intensive training and practice to be able to solo with the Thinking Process and usually several years of applications to become proficient. Eric Noreen, Debra Smith, and James T. Mackey [26] report that even after this training, only a limited number of people are able to create significant solutions. (Their book is becoming somewhat dated, and the process, tools, and training have changed since their survey. I am not aware of more recent survey data. But, personal observations indicate that their results still apply.)

You do not need to understand all of the TOC tools to successfully apply CCPM. The reason for describing them at this point in this text is to let you know that CCPM was developed as a robust theory and subjected to extensive critical discussion before it was put to the test.

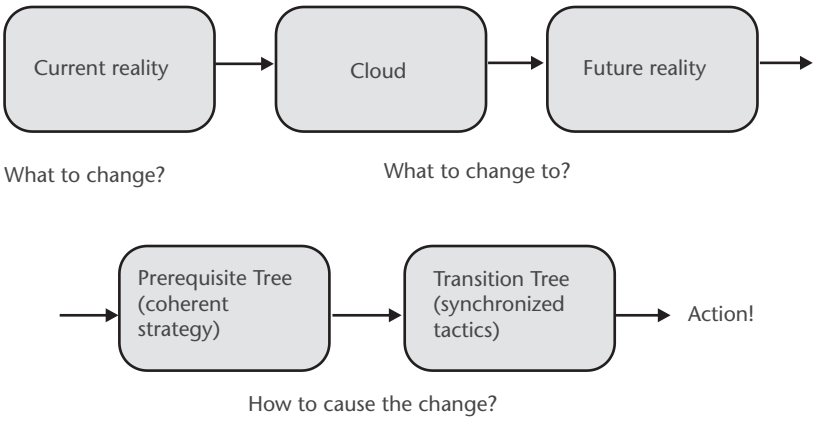


Figure 2.12 The Thinking Process leads us from UDEs, through the core conflict, to successful implementation.

Current Reality Tree The Current Reality Tree (CRT) is a logical effect→cause→effect model of the existing system connecting a core conflict to a set of UDEs. Relating all (or most) of the UDEs of the system to a single core conflict focuses on the leverage point of the system, identifying what to change. Guidelines for scrutiny (in Popper's words, "critical discussion") of the CRT leads to team agreement on the effect→cause→effect relationships that cause the system UDEs. In other words, it leads to agreement on the right problem. The CRT identifies the policies, measures, and behaviors that contribute to current reality. You subject the tree to scrojting (critical discussion) to improve it and obtain stakeholder buy-in. You read the CRT from the bottom up using IF/THEN logical statements.

Evaporating Cloud The Evaporating Cloud and guidelines for its communication and use define and aid resolution of conflicts or dilemmas. You can consider it a fixed-format horizontal tree of necessity. You read the Evaporating Cloud from left to right, using "in order to have *x*, you must first have *y*" necessity logic.

Goldratt's Evaporating Cloud can be a good tool to unearth the underlying beliefs, or mind-sets, that cause conflicts and dilemmas although unconsious beliefs are difficult to find, even with this tool.

Figure 2.13 presents a general version of the Evaporating Cloud in terms of beliefs and actions. (I have come to understand this is as the most basic representation of the Evaporating Cloud.) The cloud describes two views of reality, or two arguments (in the sense of logical arguments). Consider D and D′ as two conflicting propositions about how to achieve the goal. One argument is, "In order to have A, I must have B. In order to have B, I must have D." The other argument is, "In order to have A, I must have C. In order to have C, I must have D′." Thus, even with a common goal, there are two logical ways to get there. The beliefs may be compatible with each other, or they may not. The actions are not compatible. If they were, there would not be a conflict.

The process to resolve the Evaporating Cloud is as important as the construct. Usually one side constructs the cloud, with a pretty clear view of what the alternative actions are (e.g., D, D′). The constructor can usually come up with a belief that connects his proposition to the goal. He can only guess at the other side's belief. (As noted above, neither side may really understand his underlying belief.) The constructor presents the cloud to the other side, reading the other side first. When reading it, the constructor makes clear that he only guessed at C and accepts any revision proposed by the other side. The constructor then reads his side, noting, "No wonder we have a disagreement." He then suggests, "Let's search for solutions that will give us

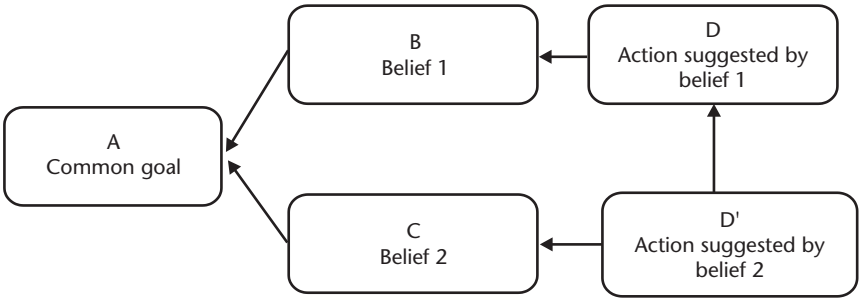


Figure 2.13 Goldratt's Evaporating Cloud provides a tool to resolve conflict and dilemmas.

A, B, and C, and not worry about D and D′. This is a win–win solution. Let's try to identify some assumptions that underlie the arrows in this diagram, and see if we can come up with a way to invalidate one or more of those assumptions, and get to our win–win solution."

Future Reality Tree The Future Reality Tree (FRT) defines the system you wish to change to. The FRT system converts all of the UDEs of current reality into their counterpart DEs. It identifies the changes that you have to make in current reality to cause the DEs and provides the effect→cause→effect logic from those changes to the DEs. The FRT identifies the feedback necessary to maintain the future reality after the actions made to create the injections are no longer active. You must scrutinize the FRTW with stakeholders of the system. You read the FRT from the bottom up using IF/THEN logical statements.

Negative Branch The Negative Branch (NBR) aids diagnosis and resolution of single UDEs. It is a tool to identify and eliminate or correct potential unintended consequences from the changes you make in the system. It is a little tree (thus a branch) connecting some known effect to the UDE. Guidelines for scrutiny and buy-in by affected people are the same as for the CRT and FRT. When used in the Thinking Process, the NBR starts by assuming successful application of one of the injections made to create the FRT. You read the NBR from the bottom up using IF/THEN logical statements.

Prerequisite Tree The Prerequisite Tree (PRT) provides a coherent strategy and synchronized logical plan to achieve a team objective. It creates team buy-in to the intermediate objectives necessary to reach a higher-level objective, such as an injection on the FRT. It makes use of people's natural ability to identify obstacles to achieving objectives and creates a logical, sequenced plan to overcome all of the obstacles. You read the PRT from the top down, using "in order to have x, we must first have y" necessity logic.

Transition Tree The Transition Tree (TRT) provides clear instructions for actions to achieve the objectives specified on a PRT or any other objective. It is a logical way to write an effective procedure. The TRT specifies the actions, the reason the action is needed, the result expected from the action, the logic for expecting the action to create the desired result, and the logic for the sequence of the actions. You read the TRT from the bottom up using IF/THEN logical statements. I have found that most TOC practitioners use the TRT very little and personally substitute project plans for this purpose.

2.7 Change Management

Implementing new management theories always requires changing the way people behave. This requires changing the system that reinforces current behavior to a system that reinforces the new desired behavior. Although many ascribe personal motivations to this resistance to change, and some apply models based on cognitive approaches, I have come to believe that Skinner's model is most effective in understanding what is going on and how to cause the necessary changes in behavior. L. Braksick [27] provides a research-based approach effective in the business environment. Her approach, based on the simple antecedent-behavior-consequence (ABC)

model of operant conditioning, has similarities to the Six Sigma approach to process analysis and improvement.

J. Kotter [28] describes an organizational approach to change management that many TOC practitioners have successfully applied. Kotter's model uses the following eight steps (p. 7):

1. Establish a sense of urgency.
2. Form a powerful guiding coalition.
3. Create a vision.
4. Communicate the vision.
5. Empower others to act on the vision.
6. Plan for and create short-term wins.
7. Consolidate improvements while producing more change.
8. Institutionalize the new approaches.

I have found this general approach useful in planning the change to CCPM, but I must caution you that actual resistance to change comes with a real face and personality. You must prepare to deal with it at the personal level. For that purpose, Braksick's approach provides the best guidance I have found.

Goldratt proposed a "layer of resistance" model, used by many TOC practitioners. While this model has substantial cognitive appeal, I have not found it useful to bring about change. My experience indicates that Kotter's model applied at the organizational level, supplemented by Braksick's approach at the individual level, produces a more effective, action-oriented approach. Chapter 9 describes an effective implementation approach using these models.

2.8 The Grand Synthesis

Although each of the management theories devolves into considerable detailed complexity, two perspectives can bring the observed results into perspective. First, TOC provides a strategy and tool to focus any other management approach on the constraint. One can easily see how one company could have great success with TQM, Six Sigma, or Lean, and another very little success, simply by focusing on the constraint, or not. If you focus on something other than the constraint, you will invest the same effort but see little bottom-line result. I believe this to be the primary difference between those who succeed with any new management approach and those who do not. I think many have succeeded by stumbling onto the constraint, while others have not been so lucky. For this reason, I do not find that critical thinking or data supports either of the premises inherent in the title of P. Pande, R. Neuman, and R. Cavanagh's Chapter 3 [10], "Why Is Six Sigma Succeeding Where Total Quality Failed?" Both succeed when applied to the constraint. The effectiveness of change management provides a secondary determinant of success. Many companies that attempt TOC also fail to see significant improvement and/or lose the gains they have made after a few years for this secondary reason.

The second perspective is that all of the management theories are accessible through the perspectives of Deming's System of Profound Knowledge.

2.9 Summary

This chapter described how thinking from four related management disciplines combines to improve the generic system for project management. There is considerable overlap between these disciplines and little disagreement on fundamental values and principles. I hope you agree after reading this chapter that

- The PMBOK™ Guide describes a comprehensive project system (present theory).
- The principles and practices of TOC provide tools to improve the theory.
- Perspectives from Lean and Six Sigma help you understand how best to apply TOC to the PMBOK™ domain.
- Lean, Six Sigma, and TOC all operate with Deming's points of profound knowledge: appreciation for a system, understanding of variation, a theory of knowledge, and understanding of psychology.
- The PMBOK™ Guide and much of its supporting literature do not differentiate between special-cause variation and common-cause variation. Given the topological similarity of the project system to the production system (Chapter 1), and given that the TOC solution vastly improved production systems, a similar TOC solution to the project system may remove many present UDEs.
- TOC provides a logical process to improve a system to determine "what to change," "what to change to," and "how to cause the change."
- The TOC five focusing steps provide the steps to implement the improvement process: identify the constraint, exploit the constraint, subordinate everything else to the constraint, elevate the constraint, and do not let inertia prevent further improvement.
- Improvement to the project system must first identify the system constraint (core conflict) leading to the UDEs of the present project system (or current theory). The core conflict will identify what to change.
- The TOC Thinking Process leads to the new system design, or "what to change to."
- Change-management approaches are necessary to implement the degree of behavior change necessary to achieve the results promised by CCPM.

The problem definition in Chapter 1 and the theory background in this chapter have set the stage to develop an improved theory for project planning and control. TOC provides strategy to apply the tools of Lean and a Six Sigma for that purpose.

References

[1] PMI, *A Guide to the Project Management Body of Knowledge*, Newton Square, PA: PMI, 2000.

[2] PMI, *Organizational Project Management Maturity Model Knowledge Foundation*, Newton Square, PA: PMI, 2003.

[3] Carnegie Mellon University, *The Capability Maturity Model: Guidelines for Improving the Software Process*, Reading, MA: Addison-Wesley, 1994.

[4] Womack, J., D. Jones, and D. Roos, *The Machine That Changed the World: The Story of Lean Production*, New York: HarperCollins Publishers, 1990.

[5] Womack, J., and D. Jones, *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, New York: Simon & Schuster, 1996.

[6] Dettmer, W., *Beyond Lean Manufacturing: Combining Lean and the Theory of Constraints for Higher Performance*, 2000, available at http://www.goalsys.com/HTMLobj-786/TOC_and_Lean_Paper__Dettmer-rev_.pdf (accessed April 30, 2004).

[7] Wikipedia, "Agile software development," available at http://en.wikipedia.org/wiki/Agile_Methods (accessed June 21, 2004).

[8] Anderson, David J., *Agile Management for Software Engineering*, Upper Saddle River, NJ: Prentice Hall, 2003.

[9] NIST, *Baldrige, Six Sigma, & ISO: Understanding Your Options,* 2002, available at http://www.quality.nist.gov/PDF_files/Issue_Sheet_SS.pdf (accessed April 30, 2004).

[10] Pande, P., R. Neuman, and R. Cavanagh, *The Six Sigma Way: How GE, Motorola, and Other Top Companies Are Honing Their Performance,* New York: McGraw-Hill, 2000.

[11] Hendricks, Kevin B., and Vinod R. Singhal, "Don't Count TQM Out . . . Evidence Shows Implementation Pays Off in a Big Way," *Quality Progress*, April 1999.

[12] Deming, W. Edwards, *Out of the Crisis*, Cambridge, MA: MIT Press, 1982.

[13] Deming, W. Edwards, *The New Economics*, Cambridge, MA: MIT Press, 1993.

[14] Senge, Peter, *The Fifth Discipline*, New York: Doubleday, 1990.

[15] Brooks, Frederick P., *The Mythical Man Month: Essays on Software Engineering,* Reading, MA: Addison-Wesley, 1995.

[16] Hardin, Garrett, *Filters against Folly*, New York: Viking, 1985.

[17] Popper, Karl R., *Objective Knowledge, An Evolutionary Approach*, Oxford: Clarendon Press, 1979.

[18] Shewhart, Walter A., *Statistical Method*, New York: Dover, 1986.

[19] Skinner, B. F., *Science and Human Behavior*, London: The Free Press, Collier Macmillan, 1953.

[20] Kohn, Alfie, *Punished by Rewards,* Boston: Houghton Mifflin, 1993.

[21] Herzberg, Frederick, *Work and the Nature of Man,* Cleveland, OH: World Publishing, 1966.

[22] Goldratt, Eliyahu M., *Theory of Constraints*, Croton-on-Hudson, NY: North River Press, 1990.

[23] Dettmer, H. William, *Eliyahu M. Goldratt's The Theory of Constraints, A Systems Approach to Continuous Improvement*, Milwaukee, Wisconsin: ASQC Press, 1997.

[24] Goldratt, Eliyahu M., *The Goal*, Great Barrington, MA: North River Press, 1984.

[25] Goldratt, Eliyahu M., *It's Not Luck*, Great Barrington, MA: North River Press, 1994.

[26] Noreen, Eric, Debra Smith, and James T. Mackey, *The Theory of Constraints and Its Implications for Management Accounting*, Great Barrington, MA: North River Press, 1995.

[27] Braksick, L. *Unlock Behavior, Unleash Profits*, New York: McGraw-Hill, 2000.

[28] Kotter, J. "Leading Change: Why Transformation Efforts Fail." In *Harvard Business Review on Change,* pp. 1–20, Boston, MA: Harvard Business Review Publishing, 1998.

CHAPTER 3

The Direction of the Solution

3.1 Deciding What to Change

The most important decision you make when you go about improving anything is determining what to change. Everything else follows from that decision. If you decide to change something that is not the constraint of the system, you most likely will not affect the system. You could make it worse by making a new constraint more restrictive than the old constraint. But you can never make the system better by improving a nonconstraint.

Throughout my career I have witnessed dozens of organizational structure changes, all attempting to improve the performance of the organization. None of them ever did. I have also witnessed several attempts to improve project management through improved software, more training, or more procedures that failed to achieve significant performance improvement. In each case, the physical change was accomplished—boxes added onto the organization chart, people trained, software purchased (and, in some cases, used), books of procedures written—but project performance remained about the same. Of course, managers changed as well. TOC taught me that this means one thing: the solution did not exploit the system constraint. The one experience I had before learning of TOC that did result in significant change did, in retrospect, address the constraint. It included many features of CCPM. At the time, the leaders of the change did not understand the theory underlying CCPM. If they had, the change would have been even more successful.

Now, having seen CCPM deliver in multiple organizations, I know why.

3.1.1 Defining the Project-Management System

The goal or aim of the project system is to deliver project results that satisfy all project stakeholders. This requires delivering the promised scope, on or before the promised delivery date, at or under the estimated cost. The black-box view of the project system (Figure 3.1) clarifies the system goal, identifies the system inputs and outputs, and leads to the measures that aid controlling the system to achieve the goal.

3.1.2 Project Failure as the Undesired Effects

The theory of knowledge leads us to define a new problem to improve the project system. Comparing predictions of the current project system (the theory) with reality helps to define the problem. UDEs differ from the DEs necessary to support the goal of successful projects. UDEs identified in Chapter 1 include the following:

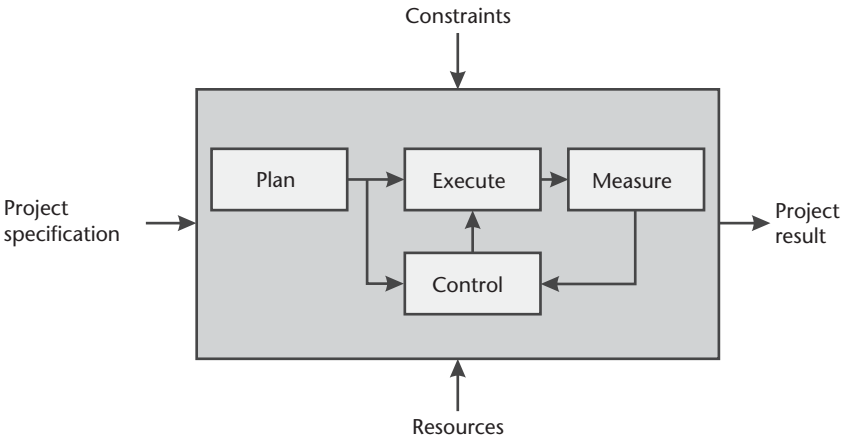


Figure 3.1 The black-box view of the project-system processes inputs to produce outputs that satisfy the system goal.

1. Projects frequently overrun schedule.
2. Projects frequently overrun budget.
3. Projects frequently have to compromise on scope to deliver on time and budget.
4. Projects have too many changes.
5. In a multiproject company, projects frequently fight over resources.
6. Project durations get longer and longer.
7. Many projects are cancelled before they complete.
8. Project work creates high stress for many participants.

UDEs are things that we do not like about the present system. A good way to check that these are indeed UDEs is to state them with the lead-in, "It really bothers me that…" Your UDE list may not include some of these, and it may include others. Feel free to add or delete as necessary.

TOC helps us to understand that these effects are a direct result of the project system we are currently using. Even though they are not intended effects, persistence of the UDEs for some time demonstrates that they are robust effects of the system. This means that there are things elsewhere in the system that cause the UDEs. Since the UDEs are observed in all types of projects in all types of businesses in many types of cultures, we can conclude that project type, business type, and culture are not primary factors or influences that cause these results. TOC leads us to suspect that an underlying conflict or dilemma common to all of these environments causes these effects. In order to decide what to change, you first have to identify this dilemma: the constraint of the present system.

3.2 Identify the Constraint

Chapter 1 asserts that any project worth doing is worth doing fast. The reason is that, for most projects, investment starts when the project starts, but return on investment does not start until the project is complete. Thus, the goal of a started project should be to finish ASAP. We will consider the constraint to this goal.

Most projects performed today use the CPM, which was developed in the early 1950s and is taught as the centerpiece of most project-management classes. (I know. I teach some at the University of Phoenix.) It is also described in all project-management texts. Figure 3.2 illustrates a typical critical-path schedule. The longest path through the network is the critical path.

Figure 3.2 also shows the resources assigned to perform each task. Assume the estimate of task duration requires the resource being fully dedicated to the task (an approach I recommend, for reasons that will become clear later on). Will this project complete on time? Not likely. The schedule plans for all of the resources to multitask (i.e., work on more than one task at a time). By so doing, they will extend the duration of each of the tasks they are working on and, thus, of the overall project. Since the project will be late if only one task on the critical path takes longer than planned, this project is planned to be late. This is true for almost all CPM-planned projects as nearly none of them have infinite resources.

Analyzing the project illustrated in Figure 3.2 allows us to estimate how long it might actually take. Consider the work performed by resource 1 (tasks 1, 3, and 5). Since all three tasks are planned to start at the same time, this means each task will take three times as long while the three are worked simultaneously. Thus, as the shortest task (task 1) is 5 days long, after 15 days, task 1 will be complete, and tasks 3 and 5 will have five days of work performed on them. Then, resource 1 must split time between the two remaining tasks, meaning each task progresses at the rate of one day process per two days of elasped time. Therefore, the remaining 20 days of work on task 2 will take 40 days, and the total actual duration for task 3 will be 55 days. After 55 days, task 5 still has 15 days of work on it, so its total duration will be 70 days.

Figuring out how long resource 2 will take on each task gets more complicated because of the task dependencies. Resource 2 can start on task 2 with 100% effort on day 15 and will not be able to start on task 4 until resource 1 completes task 3 (i.e., until day 55). Thus, resource 2 can work 40 days focused on task 2, but must split time for the final 5 days of work with task 4, causing the duration of task 2 to increase to 50 days. Task 4 is now overlapping with task 2 and task 6, extending its duration to 40 days. Figure 3.3 illustrates the expected actual performance of the project, extending the date by over a month. The project was planned to fail.

ID	Task Name	Duration	Predecessors	Resource Names
1	Task 1	5 days		1
2	Task 2	45 days	1	2
3	Task 3	25 days		1
4	Task 4	30 days	3	2
5	Task 5	60 days		1
6	Task 6	5 days		2
7	Project End	0 days	2, 4, 6	

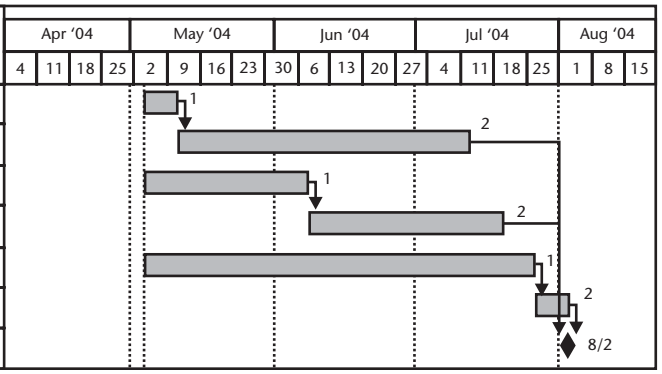


Figure 3.2 Example critical-path schedule.

ID	Task Name	Duration	Predecessors	Resource Names
1	Task 1	15 days		1
2	Task 2	50 days	1	2
3	Task 3	55 days		1
4	Task 4	40 days	3	2
5	Task 5	70 days		1
6	Task 6	10 days	5	2
7	Project End	0 days	2, 4, 6	

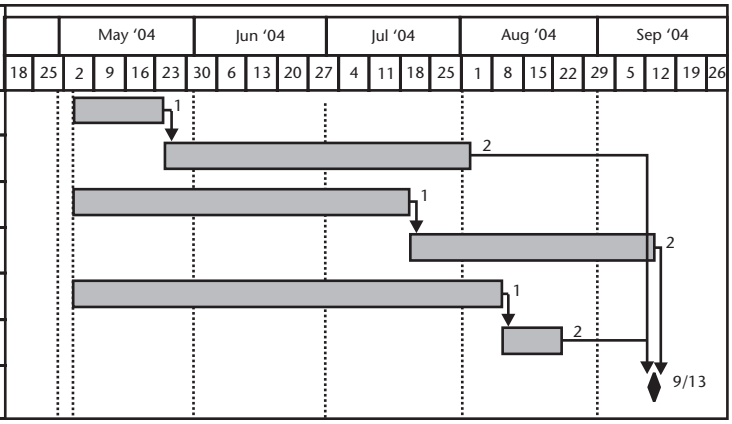


Figure 3.3 Actual task performance.

A method exists to solve this problem: resource leveling. Most CPM software includes the ability to resource level. Figure 3.4 illustrates the resource-leveled version of Figure 3.1. Note that the date of the resource-leveled schedule conforms to the date of the actual schedule. It also resolves the first UDE: projects frequently overrun schedule. Of course they do: they are planned to!

Examination of Figure 3.2 and 3.4 leads to an interesting observation. The software identified the critical path as only comprising tasks 5 and 6. This is curious as there is no critical path for the upper path of the project. Just why it chose this is a mystery to me, and I know that the identified critical path after resource leveling will be different with different software, even if it does the leveling the same way. The reason is that the critical path is not defined after resource leveling. Before resource leveling, the critical path has no extra time in it (float or slack). Notice on Figure 3.2 that the two noncritical paths, the path through tasks 1 and 2, and the path through tasks 3 and 4, each have extra time, shown by the line with no task bar. After resource leveling (Figure 3.4), all of the paths have float (or slack). Thus, none is the critical path.

I do an informal survey in the classes I teach for the PMI. I ask how many students resource-load (i.e., identify the resources needed, as in Figure 3.2) their project

ID	Task Name	Duration	Predecessors	Resource Names
1	Task 1	5 days		1
2	Task 2	45 days	1	2
3	Task 3	25 days		1
4	Task 4	30 days	3	2
5	Task 5	60 days		1
6	Task 6	5 days	5	2
7	Project End	0 days	2, 4, 6	

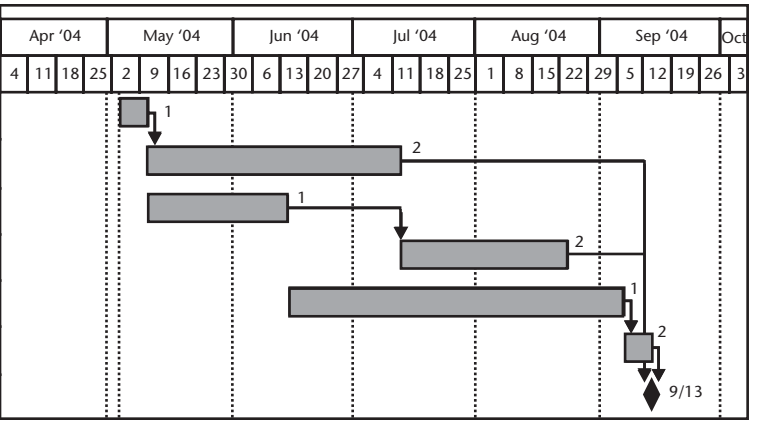


Figure 3.4 Resource-leveled critical-path schedule.

plans. Usually one half to two thirds do so. I then ask how many resource-level their plans. Usually, it is about 5%. Thus, about 95% of projects are planned to fail in the first place. Keep in mind this is a group made up of the elite in project management: most are certified PMPs.

I ask those that resource-load but do not resource-level why they work this way. Most do not have an answer, but those who do say one of two things:

1. The date moves out (!).
2. The leveling leads to sequences that do not make sense.

The first point indicates a reluctance to deal with reality. I will deal with the second point later.

For now, we are going to follow Dr. Eliyahu Goldratt and suggest a simple definition change: the constraint to a single project is the critical chain, defined as the longest path through the network after resource leveling. Figure 3.5 illustrates the critical chain for our simple example project. Note that it has no float or slack when defined. Note also that it jumps the project logic paths (although it retains the technical logic paths in the project plan.)

In the past, I never questioned the proposition that an acceptable way to remove resource contention is first to identify the critical path and then level resources. My literature search did not reveal the basis for this proposition. I suspect, but have no proof, that this may be a result of technological evolution. It is possible to calculate a project manually to find the critical path. There is no simple algorithm to create an optimum, resource-leveled critical path. Thus, it is very difficult to resource-level even a modestly complex project plan manually. The relatively expensive and slow computers that existed at the time of the growth of CPM and PERT did not lend themselves to doing a lot of calculation. The idea that you could use the computer to calculate the critical path, lay out the network, and then deal with the potential resource constraint seems logical enough. It may even have been that for the projects using CPM and PERT, resources were less often a constraint. They could find the critical path and then determine and satisfy the resource demand.

ID	Task Name	Duration	Predecessors	Resource Names
1	Task 1	5 days		1
2	Task 2	45 days	1	2
3	Task 3	25 days		1
4	Task 4	30 days	3	2
5	Task 5	60 days		1
6	Task 6	5 days	5	2
7	Project End	0 days	2, 4, 6	

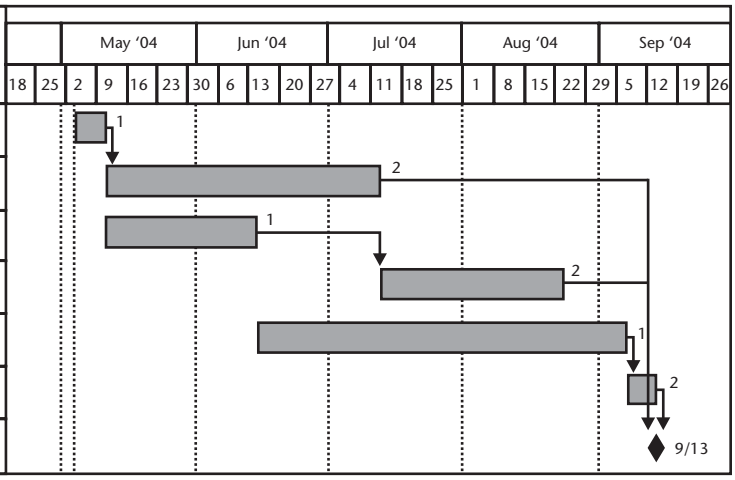


Figure 3.5 The critical chain is the longest path through the project after resource leveling.

Current project-management software operates by starting with the activity structure (critical path) and only then considers the limited resources available for the project. Project-management software identifies the critical path by linking the project activities in a logical way, then measuring the longest time through the network of activities, assuming no resource constraints. The project manager inputs resource availability. The software then allocates the resources through various schemes, but usually first to the critical path (i.e., by least float or slack) and then to the paths that are nearest to the critical path in time duration (minimum-slack activities first). People who have studied resource allocation know that this does not always give the optimum schedule. People have proposed various heuristics, and some programs provide a large number of selections. The only way to find the optimum among these options is trial and error. Most software gives you some degree of control over the process.

Thus, Goldratt's first key insight is to identify the constraint of a single project as the critical chain, instead of the critical path. The critical chain includes *both* task and resource constraint.

3.3 Exploit the Constraint

The original TOC Thinking Process method went directly from the UDEs to create a CRT, a system model of the present reality that was causing the UDEs. The procedure started with any two UDEs and built a logical connection between them. It then added one UDE at a time, filling in the logic until all of the UDEs were connected in a system representation of current reality. After a process of what Karl Popper would call "critical discussion," the analyst would select an entity in the tree causing all or most of the UDEs as the core problem, and proceed to analyze it as the result of a conflict. This led to an initial change to begin the design of a new system, which no longer created the UDEs and, in fact, created their opposing DEs. This process worked. However, it was hard and lengthy.

A recent innovation, which Goldratt indicated was suggested to him by someone else (but he did not say by whom), made the process more direct and seemingly easier to operate by more people. This method selects three of the UDEs and analyzes each of them as stemming from a conflict. It then considers the three conflicts together to define an underlying core conflict. Finally, the revised method uses the core conflict to construct the model of current reality, showing how the core conflict leads to all (or most) of the UDEs in the system. This process concludes with identification of the initial change necessary to begin to revise the system to generate a future reality free of the UDEs. While I have reservations about using this process when analyzing a new problem, I find it a useful way to describe a completed analysis and, thus, will use it to address how to exploit the critical chain of a project.

3.3.1 Projects' Durations Get Longer and Longer

Most people agree that projects seem to take longer and longer.

I ask people in classes, "Does everyone know what *contingency* is?" All participants usually signal that they do indeed understand it. Then I ask someone to define it. A lot of wiggling in place usually follows the question, but eventually, sometimes

after I single out an individual, someone offers an answer along the lines of, "Extra time or money to handle the unexpected." I then ask, "Extra compared to what?" More puzzled expressions. I refer to Figure 3.6 as an example of the variation in task performance (which they have previously experienced by an estimating exercise), and ask, "Isn't it a huge difference if you add contingency to the 50% probable task estimates, as compared to adding it to the 90% probable task estimate?" They all agree and understand that the word *contingency* can have a vast difference in meaning, depending on how you choose to interpret the base. I offer an operational definition: "Contingency is the difference between a 50% probable estimate and a 90% probable estimate." If you do not like this definition, you are welcome to change it. Just be sure that the people you are communicating with are using the same meaning.

Everyone wants to have a successful project. One necessary condition for this is to have the project complete on schedule. In order have projects completed on schedule, we must have every task on the critical path complete on schedule. In order to have every task on the critical path complete on schedule, we must plan each task to include the contingency (as defined above) because we know that there is uncertainty in task performance. This is the only way to do it with the present CPM. Further, since you only find out the critical path by estimating all of the project tasks and connecting the network, you have to include contingency in all of your task estimates.

Project managers generally agree that they want people to keep their commitments and deliver on their task delivery date. People generally agree that in their organizations, people who complete tasks on time are good performers, and people who do not complete tasks on time are considered poor performers. They acknowledge that when project managers ask for input on task times, they want contingency included in the estimates.

Usually, there is also pressure to plan to complete projects as soon as possible. In competitive bid situations, the bidder who can complete sooner usually has an edge. Everyone knows that planning to complete the project sooner tends to reduce project cost, therefore helping to make a competitive bid. For those performing R&D projects, the impact of shorter development may make the difference between

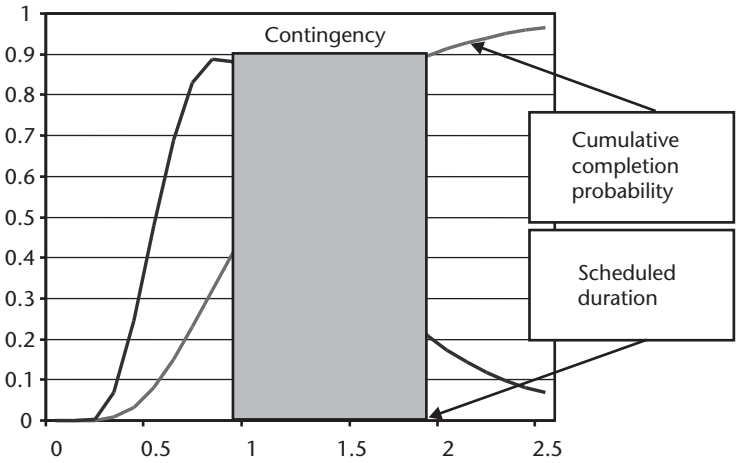


Figure 3.6 Variation in estimates for the time to perform a task helps define contingency.

the success and failure of the project. For deadline-driven projects, a shorter plan time usually alleviates the pressure to start now.

For all of these reasons, in order to plan a successful project, the project manager must have a shorter critical path for the project. In order to have a shorter critical path for the project, the project manager must have shorter task estimates that do not include contingency.

Figure 3.7 shows the Evaporating Cloud for the dilemma we describe above. Of course, we cannot have both 50% probable task estimates and high-probability task estimates, so there is a conflict. In many environments, this conflict plays out with the task estimators proposing high-probability estimates and management, including the project manager, reducing these estimates as a "challenge" or "stretch" goal. These time cuts usually do not include a method to achieve the time reduction. They are arbitrary. Usually people know that management still expects them to achieve these low-probability task times. They go into the schedule as fixed dates, and management will request status to that date.

Task performers tend to accept the "challenge." They really have no option. There is considerable pressure to be a team player and to do your part. Subcontractors often face the same pressure; meet the reduced time, or we will give the work to someone else. Experienced people justify accepting the situation as a management-dictated version of the chicken game. Remember the old movies where two drivers would race toward a cliff or toward each other to see who would "chicken out" first and veer off or stop the racing car? People on a project know that what is happening to them is also happening to every other task on the project. If they agree to the time cut, it is very likely that reality will strike some other project task before it hits them, causing management to chicken out and extend the project time. This will give them the time they need to complete their task, so they can win in the system. If they were to object to the time cut, they would lose immediately, as management would brand them "nonperformers" or "nonsupporters." They have no choice in the real world of power politics.

3.3.2 Projects Frequently Overrun Schedule

When asked why projects overrun schedule, people usually say the projects start out fine, but somewhere along the way, a snag develops that begins to push one or more deliverables farther and farther behind. Everyone knows that it only takes one late

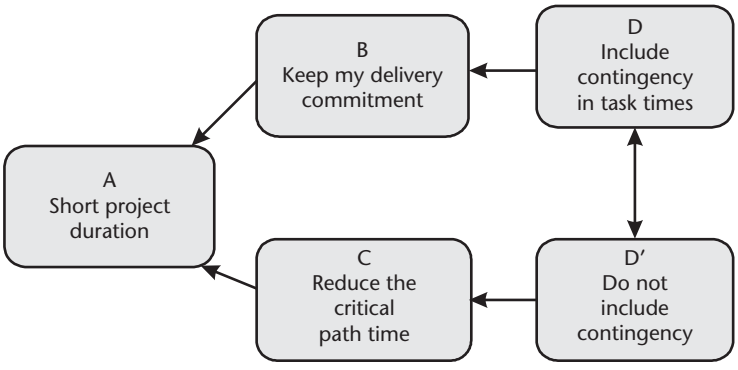


Figure 3.7 Task time conflict.

task on the critical path to make the whole project late. As this shift begins to hit the plan, management tries to solve the problem causing the shift, usually by diverting resources and making changes in the project plan, placing more and more emphasis on the part of the plan that is slipping. The people working on the snag usually feel a lot of pressure to get their part of the project solved and, therefore, put in a lot of extra time and feel considerable stress. These are often the resources in most demand in the company, so putting more time on the project in trouble leads them to neglect the other projects they are supposed to be working on, causing other projects to slip as well.

When asked why this happens, people respond with two general types of answers. One type of answer focuses on the specific problem with the specific project that is most recent in memory (often still in trouble). They usually blame the slippage on poor performance by the group responsible for that part of the project. The second type of response is more general, blaming the slippage on the tendency of stereotypical task performers to underestimate or on management's setting arbitrary completion dates.

How often do people complete activities and pass their work on early? How often do they complete activities for less than the budgeted activity cost? You might find this occurs less frequently than you expect, if the estimates are truly 90% probable estimates. Even with skewed distributions, tasks should complete early a substantial percentage of the time. Figure 3.8 illustrates typical results for the actual times that project tasks take as compared to their planned duration. Most tasks complete exactly on the due date; often as many as 80% complete on the due date. This is not consistent with the task-completion-time estimate presented earlier.

Potential causes for little positive variation in activity duration or cost include the following:

1. People work diligently to milestone dates and do not understand a desire to have the work completed early.
2. Estimates are much less probable than we believed, leaving little potential for positive variations.
3. The work expands to fill all available time and budget.
4. There is a belief that the next activity will not be ready to use the work anyway.
5. In most organizations, there are significant penalties (threatened or real) for completing activities late. What are the rewards for completing activities early?
6. Reduced budget for the performing organization leads to higher overhead rates and, in extreme cases, downsizing.
7. Reduced credibility of performer's activity duration and cost estimates leads to increased pressure to reduce the estimates.

These factors add to the psychological reasons that cause projects to lose much potential positive variance. Project managers assign tasks and train people to respond to specific milestone dates. Thus, even if they are done early, they might hold on to the product until the due date. Why not? Management usually does not take advantage of early completion or reward the task performers if they do deliver

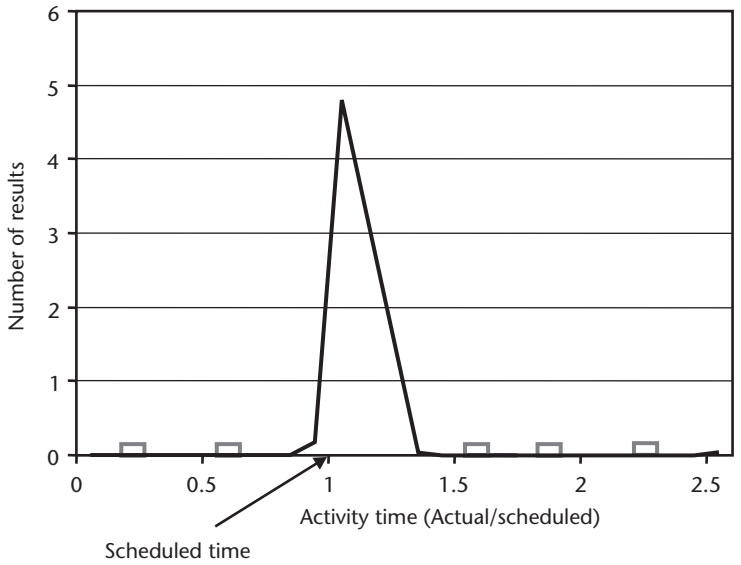


Figure 3.8 The distribution of actual task completion time differs from the estimate distribution. It shows a remarkable percentage of completions right on the due date.

early. Paying the resource performing the work in accordance with the time they spend on the task motivates him or her to use all of the resource authorized. Using a "cost-reimbursement" contract with the resource (the usual practice for resources in the company and for certain types of external resources) may provide him or her incentive to slow down the work to get overtime pay or more total revenue from the project.

If one resource gets an activity done early, what is the chance that the next critical resource down the line is ready to hop to and start working on his or her activity? If it is a critical resource, it is in demand and has limited availability. It does not seem likely that resources will be able to work on activities until the scheduled dates. Therefore, we lose the positive variance and introduce wait time. This means that the actual schedule time grows due to activity dependence.

All of this leads to the second conflict illustrated by Figure 3.9. The upper path refers to the performing resource. In order to be a successful team member, I must

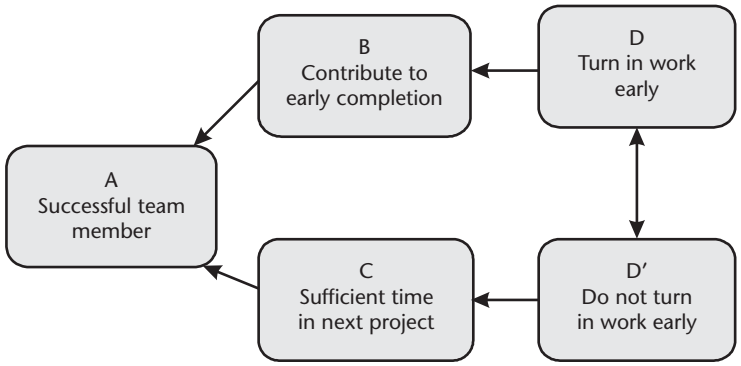


Figure 3.9 The conflict underlying project schedule overruns.

contribute to early completion of the project. In order to contribute to early completion of the project, I must turn work in early. On the lower branch, in order to be a successful team member, I must have sufficient time in my task estimates to complete my commitments. In order to have sufficient time to complete my commitments, I must not turn tasks in early. The obvious answer to this is that I can always do extra checks and improve the quality of my project-task result when it looks like I might finish early. Even if I do finish early and turn the work in to my manager to be checked prior to submitting it to the project, she will not likely look at it until it is due anyway as she is a very busy person.

Student Syndrome Did you always study for your exams weeks ahead so that you could go to bed early the night before? Did you always write your papers to get them done at least a week before the deadline to avoid the gap in the library where all the books on the topic used to be and to get to the college computers before everyone else was on them all night? (They did not have computers when I was in college, so this was not a problem for me.) Are you normal?

Well, it is probably not news to you that you are normal and that most people have a tendency to wait until tasks get really urgent before they work on them. This is especially true for busy people in high demand, that is, all of the most important people that the project manager is counting on to get the critical-path work done on time.

Figure 3.10 shows the typical work pattern of many people. They do less than a third of the work on an activity during the first two thirds of the activity duration. They do two thirds of the work during the last third of the activity duration. Where are they more likely to find they will have a problem completing the activity in the remaining time: during the first third of the effort or during the last third? If they are working above 100% capacity already to complete two thirds of the work in one third of the time, there is no chance of keeping to the activity duration by putting in a little extra effort. What is the chance they can recover from an unanticipated problem, like a computer crash?

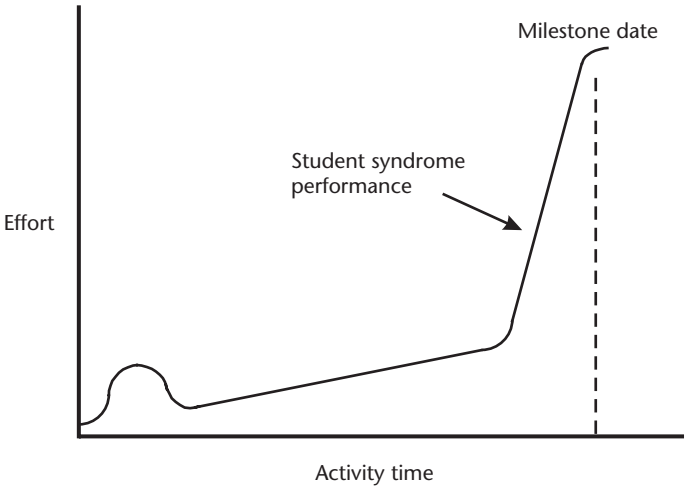


Figure 3.10 People perform most activities, and most people follow the student-syndrome performance curve.

Student-syndrome behavior results in there being little chance of seeing the positive side of activity duration variation. The effects described above make it unlikely that we could take advantage of positive variation, even if we did see it. No wonder projects rarely complete early! The reality is that relative activity duration normally shows a very skewed distribution, with a mean well above the average activity time. This is one reason why we often see overruns on activity time, but rarely see underruns.

Most project-management guidance recommends that project managers use an early-start schedule. This means they start all of the non-critical-path activities earlier than is necessary to meet the schedule date. People working on those activities know that there is slack in their activity. How do you think this influences the urgency they feel in working on the activity?

3.3.3 Multitasking

Now let us assume that the schedule system demands that workers start on activities as soon as possible and report the task start to the project manager. Let us assume the person splits his or her time during the day evenly between three activities each of which could be completed in one week with 100% effort on that task alone. When do they complete? If we assume that there is no time lost from dropping each activity every day and having to get back into it, none of the activities is complete until the third week. Multitasking has increased the activity duration for all three projects from one week to three weeks. They have delayed throughput on the first project for two weeks, and on the second for one week.

While most people will acknowledge the facts above, many argue, "It's just not realistic to do otherwise. We have to satisfy multiple needs." They agree with logic that demonstrates that multitasking is a very poor (perhaps the worst) way to meet multiple needs (Figure 3.12). They acknowledge that it deliberately lowers their personal throughput contribution. (Moreover, this still doesn't account for the fact that leaving and returning to tasks usually impacts the total time necessary to complete the task and often the quality of the product.) Nevertheless, many people find it extremely difficult to change this behavior. Purveyors of time-management tools work to resolve this conflict at the personal level.

Peter Marris [1] contends that behavior such as multitasking is a social effect of the more powerful using the less powerful to shield them from uncertainty. In other

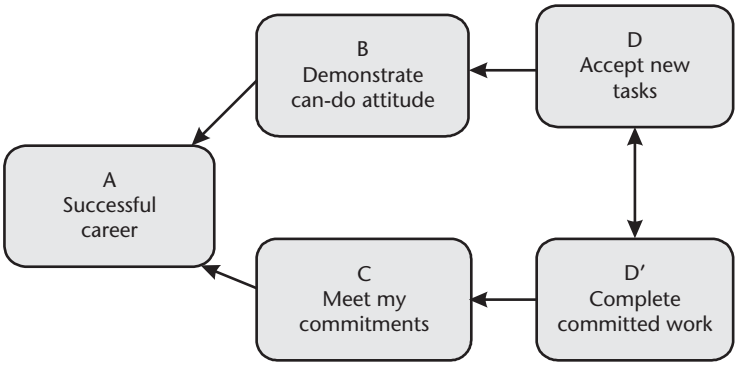


Figure 3.11 The multitasking conflict.

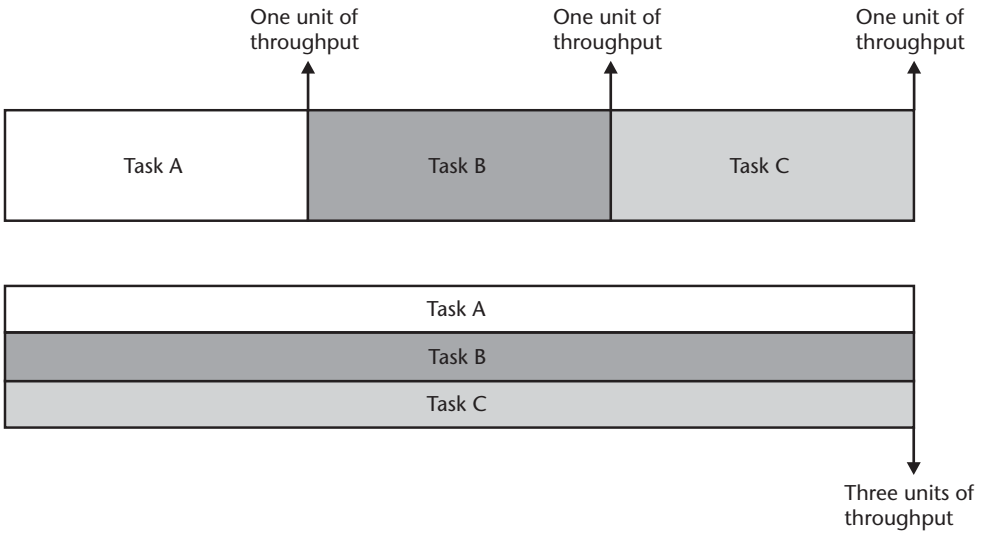


Figure 3.12 Multitasking delays all projects. It also justifies using the longer task times in future plans.

words, management takes advantage of lower-level resources in the organization by creating the pressure that leads to multitasking.

3.3.4 The Core Conflict Leads to Undesired Effects

You can combine the three conflicts to obtain the underlying core conflict leading to all three. Since the conflicts derive from the three starting UDEs, resolving the core conflict should have a desirable impact on all three of the UDEs analyzed. Since the project system is a connected system, the core conflict may contribute to the other UDEs as well.

Figure 3.13 illustrates development of the core conflict. The goal of the three conflicts is common: project success. The top path of the cloud illustrates the logic that leads each individual to work toward his/her own success. In order to have a task successful project, each task must perform as planned. In order for each task to perform as planned, each task performer must do whatever individual task success demands.

The lower path illustrates the logic that leads to working toward project success. In order to have the project succeed each part of the project must contribute to overall project success. In order to contribute to overall project success, each task must subordinate to the overall project.

This core conflict is the common conflict referred to by Deming, where working for each part of the system does not lead to an effective system. It is the conflict identified as a principle in TOC: in an optimum system, each part of the system cannot perform at optimum. Worse yet, the core conflict sets up a win–lose situation between all of the project workers and project management. No wonder projects are so stressful to all concerned. No wonder so many projects fail.

Figure 3.14 illustrates how the core conflict leads to all of the UDEs. This implies that the core conflict is a high-leverage part of the project system. A solution (new theory) that resolves the core conflict differently can influence the whole system in a way that tends to move the UDEs toward their desirable counterparts.

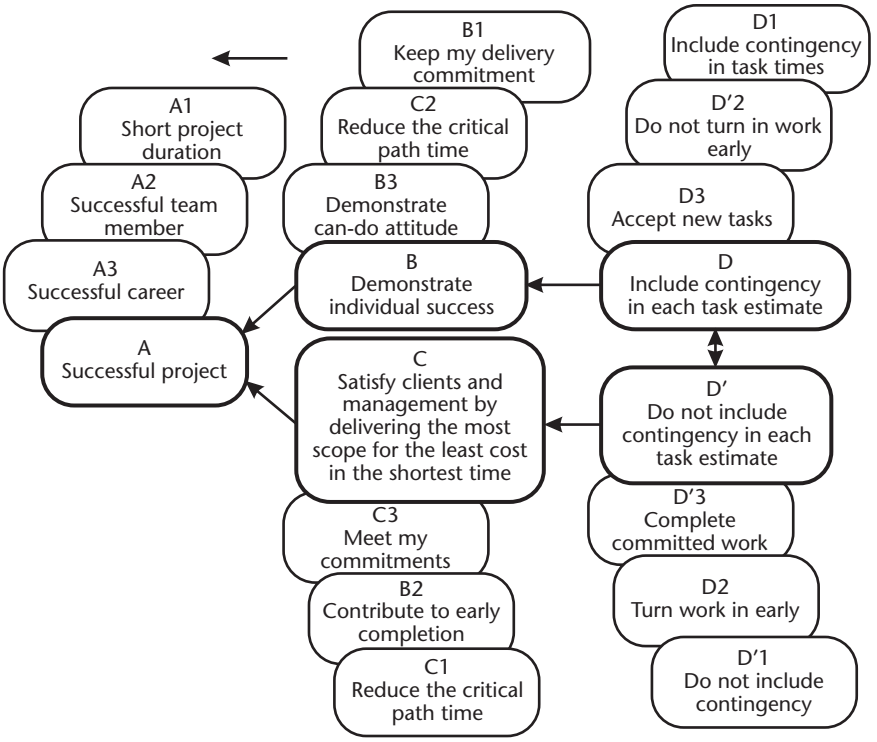


Figure 3.13 The core conflict underlies all three conflicts.

The logic illustrated by Figure 3.14 is incomplete. It is only a notional connection between some part of the core conflict and the UDE. Chapter 11 illustrates more of the complete logic underlying these connections, and you can find the complete CRT at http://www.Advanced-projects.com. At this point, if you accept that the core conflict underlies most or all of the UDEs of the project system, you may be willing to consider the beginning of the solution direction.

3.4 Toward Desired Effects

3.4.1 Resolving the Core Conflict

Resolving the core conflict requires identifying one or more assumptions that can be invalidated by changing the system. Assumptions underlie each arrow of the core conflict. The critical-chain method arises from attacking the assumption that adding contingency to each task is the only way to manage uncertainty.

Goldratt was uniquely positioned to develop the critical-chain solution for projects. The critical-chain solution comes from recognizing that the variation in task performance and dependent events is at the root of the behavior of the present system. He had tremendous success in applying the solution for production management described in *The Goal* [2]. He knew that, in most cases, the uncertainty in project-duration estimates is much larger than the variation in production processes. He also knew that in many cases the task dependencies in projects were equal to or greater than the dependencies that exist in production. It is natural that he would look at projects from this perspective to find the assumption to attack.

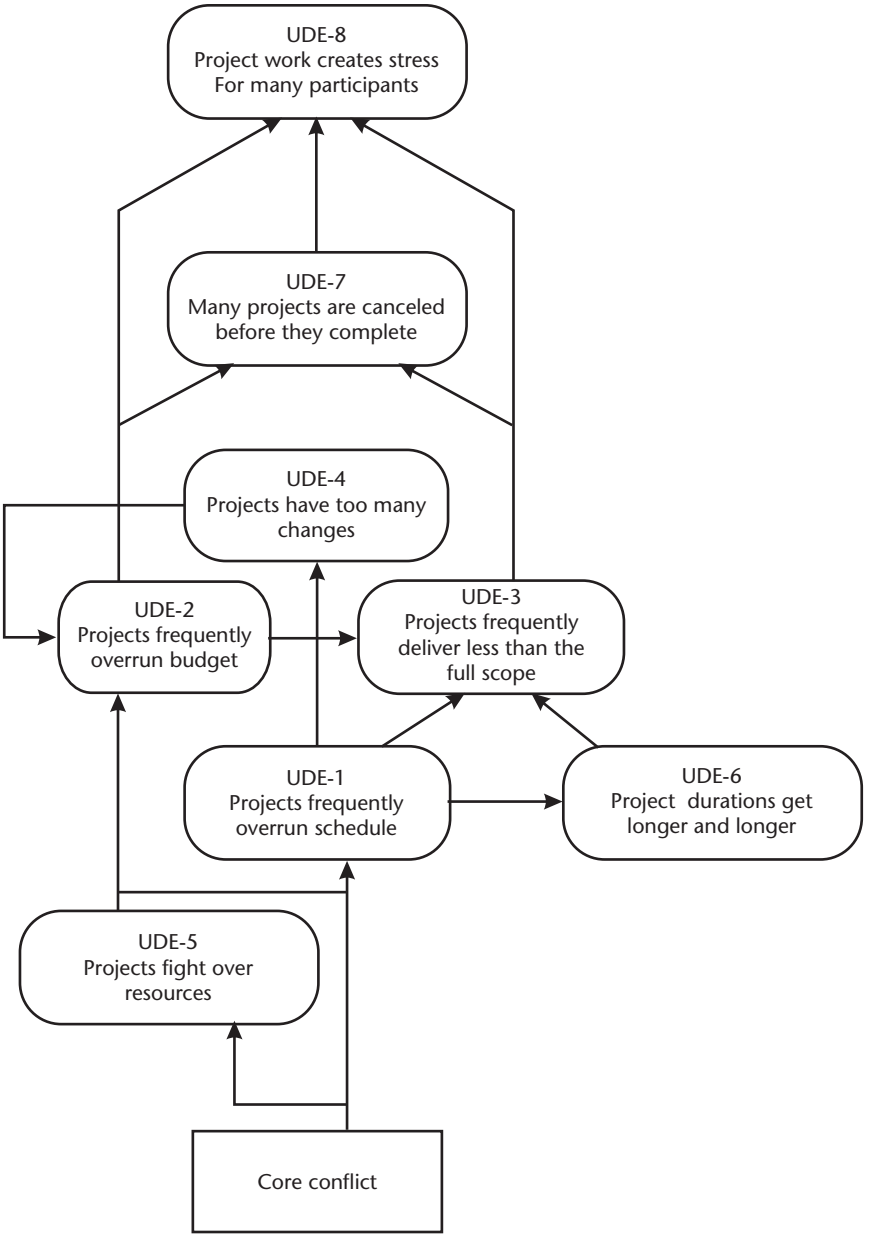


Figure 3.14 The core conflict leads to all of the system UDEs.

Goldratt describes the impact of variation and dependent events using the saga of Herbie in *The Goal*. He uses the scenario of a troop of Boy Scouts on a hike through the woods. The trail is narrow, so the scouts cannot pass one another and must walk single-file. As they hike, the line grows longer and longer. Alex Rogo, our hero in *The Goal* and the troop leader for this weekend, realizes what is happening. The speed of the individual Boy Scouts is not the same. There are statistical fluctuations in how fast they walk. Each is dependent on the Boy Scout in front of him, and the one in front of that Boy Scout, because they cannot pass each other. These fluctuations cause the length of the line to grow continuously. Herbie turns out to be the

slowest Boy Scout: the constraint. The gaps in the line compare to inventory in a manufacturing plant, which piles up while a machine is working on the parts in front of them (in the line).

For a project, the gaps in the line of Boy Scouts compare to time. If the next resource is not ready to start when a predecessor activity completes early, the project loses time. We lose the positive variances in statistical fluctuations. This is like a faster boy walking behind a slower one; he can catch up but not pass. The line grows in length. This scenario is worse than the manufacturing case. In manufacturing, eventually the inventory is used. In a project, we lose the time forever. There is no conservation of time.

The direction of Goldratt's solution follows from his TOC production solution. The first step is to identify the constraint of the project system. His focus on throughput led him to focus on the time it takes to complete the project. The longest path through the project is the evident constraint. At first look, this is the critical path.

How then to exploit the critical path? Goldratt is a Ph.D. physicist. He knows statistics, and he knows a lot about the cloudy behavior of much of reality. He knows that the only way to take advantage of our statistical knowledge is through dealing with numbers of events. Dr. W. Edwards Deming and Walter Shewhart before him had pointed out that science can't make predictions about a single instance of a statistical event. This leads to a very simple (in retrospect) insight: concentrate the uncertainty for many of the tasks of the project at the end of the project in a buffer. The buffer has a direct counterpart in Goldratt's production solution, where buffers of in-process inventory are strategically placed in front of machines to prevent them from running out of work.

Concentrating contingency in the buffer brings along two significant bonuses. The first bonus is a shorter plan. The variance of the sum of samples from a series of independent distributions is the sum of variances for the populations that samples come from. The variance is the square of the standard deviation. The standard deviation is proportional to the amount of variation in a single task. In other words, the uncertainty in the sum of tasks is the square root of the sum of the squares of the individual variation. While attempting to protect the completion date of each task in a project, each task has to include its own allowance for uncertainty. These allowances add up down the path. When we take these allowances out of each task and put them at the end of the path, they add up to the square root of the sum of the squares of the amount removed from each task; a much smaller total amount. Figure 3.15 illustrates how this works for a very simple case. The reason for this is evident.

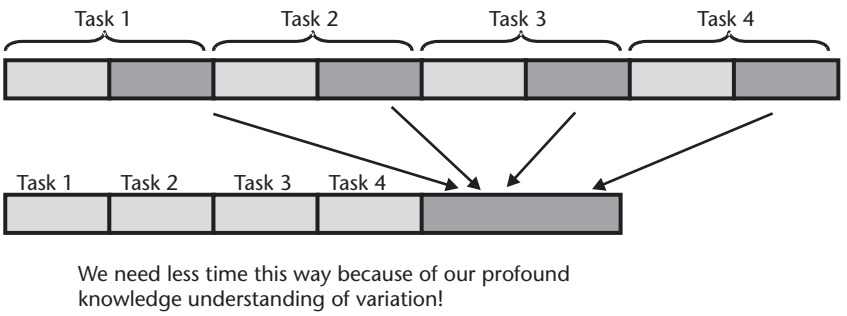


Figure 3.15 Concentrating contingency at the end of the path requires less total project time.

Some of the tasks should overrun; some should underrun. The distribution of the sum need not be as large as the sum of the individual variations because some will cancel out.

A second statistical fact comes into play with this strategy. The central-limit theorem of statistics states that the distribution of samples from a variety of independent distributions tends toward a normal distribution. A normal distribution is a symmetrical distribution. It does not have the long tail to the right that many individual task distributions may have. This means that concentrating contingency at the end of a path reduces the likelihood that it will be overrun by a large amount.

A key part of the direction of the solution Goldratt proposed, then, is to use "average" task completion times in the plan and to add an aggregated buffer at the end of the plan for overall project contingency. Mathematically, the mean for each task duration is the average duration that sums a long a path.

3.5 Solution Feasibility (Evidence)

Using the scientific method as the theory of knowledge leads to selecting the preferred theory through critical discussion and test. Comparing critical chain to critical path, we see more content in the critical-chain theory because it

1. Provides an explicit method to manage common-cause variation;
2. Explicitly resolves the resource constraint.

Popper [3] notes that a new theory should contain and explain the old. With unlimited resources, the critical chain is the same path as the critical path. With a resource constraint, the critical chain is an acceptable solution to the resource-leveled critical path. Thus, the critical chain contains the critical-path solution.

Popper suggests that the primary method of testing a new theory should be through critical discussion. This discussion checks the new theory against the old, looking for logical deductive reasoning and evidence supporting the suppositions (assumptions) made in the new and the old theory. Summarizing the reality of the scientific method, Popper states,

1. Induction (i.e., inference based on many observations) is a myth. It is neither a psychological fact, nor a fact of ordinary life, nor one of scientific procedure.
2. The actual procedure of science is to operate with conjectures: to jump to conclusions—often after one single observation.
3. Repeated observations and experiments function in science as tests of our conjectures or hypotheses (i.e., as attempted refutation).

This chapter has developed the reasoning behind the way Goldratt defined the problem with the current theory. It does not explain the jump to his proposed direction for the solution: improved management of uncertainty. It is unlikely that others without his knowledge and experience could have made the same jump. The original PERT method and subsequent work with project simulations provide evidence that others were aware of the uncertainty problem.

The current knowledge base lumps the uncertainty in predicting each project task in the area of risk management, adding evidence that people understand the need to deal with variation. However, none of the current solutions makes uncertainty management part of the basic project system in the manner of the critical-chain method. My approach to CCPM clarifies that buffers serve to manage common-cause variation, while conventional deterministic risk analysis and management handles potential special-cause variation and uncertainty. This is one way CCPM brings the power of Six Sigma and TQM to critical chain.

Critical chain explains the reasons for schedule overrun through the reality of statistical fluctuations (uncertainty or variation) and dependent events. The CPM theory does not address this reality; it uses deterministic durations, start dates, and stop dates for activities in the schedule. Combining this technical assumption with human behavior leads to schedule overrun. Schedule overrun leads to cost overrun and reduces the delivered scope. Perhaps most importantly, the new theory explains how the CPM theory, through the win–lose approach to task scheduling, causes much of the psychological harm in project systems.

The resource constraint is every bit as real as the task-input constraint. It is a necessary condition to perform the task. The CPM assumes that an acceptable solution to the resource constraint is first to find the unrestrained critical path and then to assess the impact of the resource constraint. Put another way, determination of the critical path assumes that resources are not the constraint. Alternatively, it assumes infinite resources. I could find no references describing the reasoning behind this assumption. Goldratt found it easy to notice this implicit assumption because it was made as an explicit assumption in the production system models he had worked with [3].

Some users of CCPM have asserted that the resource constraint need not be "as real" as the technical constraint. In other words, for many pieces of work, you simply cannot change the task sequence: you have to put in the foundation before you place the equipment, or you have to remove the equipment before you can reinstall it. However, often one can put resources on overtime, including extended workweeks, or elevate the resource through a variety of approaches. This does not conflict with CCPM because the plan maintains the technical sequence of tasks. All tasks require both the technical predecessor and the resource to work. CCPM does not preclude using alternatives to elevate the constraint.

This chapter has demonstrated that the CPM usually fails to identify the real constraint to the project (resources). It is a simple and logical step to define the critical chain as the combination of the two potential constraints on the longest path to complete the project.

Chapter 1 presented selected successful evidence that the critical-chain method creates the DEs. (It is selected in the sense that it is not an exhaustive listing; this does not mean we selected only the positive results!) By this time, thousands of projects of different types, in different businesses, and in different cultures around the world, have successfully applied critical chain. However, there are cases where implementation failed to achieve the changes necessary for critical chain, and the project system continued to operate the old way. Chapter 9 addresses this in detail.

3.6 Determine What to Change To

Resolving the core conflict provides a necessary change to the system to begin to move toward the DEs. The DEs for the project system, derived from the current reality UDEs, are as follows:

1. Projects always complete on or before the scheduled completion date.
2. Projects complete at or under budget.
3. Projects always deliver the full scope.
4. Projects have few changes.
5. Projects receive needed resources without internal fights.
6. Project durations get shorter and shorter.
7. All projects complete.
8. Project work creates win–win solutions for all stakeholders.

The changes to resolve the core conflict provide a method to manage uncertainty and acknowledge the reality of the resource constraints that affect many projects. The changes are not sufficient, by themselves, to create all of these DEs. While the solution to the core conflict explicitly considers the project system and addresses variation, it does not address all of the psychological elements that influence project performance. Subsequent chapters provide the complete solution leading to all of these DEs and the full solution to achieve them.

3.7 Summary

This chapter has identified the manner in which the project system *manages* uncertainty as the core conflict preventing project-management-system improvements. Specific points made in developing that theory are

- UDEs define the problem with the present project-management theory.
- We have identified the constraint to a single project as the critical chain: the resource-leveled critical path.
- We have proposed a method to exploit the constraint through better management of uncertainty, the core conflict for the present project-management system. This conflict manifests as the conflict between protecting each task in the project versus protecting the entire project.
- The project-system solution (new theory) must overcome the core conflict of the existing project by attacking a key assumption in present systems: how to manage uncertainty.
- A method to mange uncertainty using ideas similar to those that succeeded in improving production performance may provide a direction for the solution.
- The direction of the solution should be to manage uncertainty by concentrating contingency into buffers at the ends of chains of tasks.
- A growing body of empirical evidence demonstrates the feasibility of the critical-chain project-management method for all types of projects.

Note that managing uncertainty is not the same as knowing about uncertainty or analyzing uncertainty. People knew about uncertainty long before projects began. There are many methods to analyze uncertainty. Both knowledge and analysis are necessary to managing it, but they are not sufficient. You are managing only when you take actions that drive the system to the goal. Chapter 4 derives the full system to do that.

References

[1] Marris, Peter, *The Politics of Uncertainty*, London: Routledge, 1996.

[2] Goldratt, Eliyahu M., *The Goal*, Great Barrington, MA: North River Press, 1984.

[3] Popper, Karl R., *Objective Knowledge, An Evolutionary Approach*, Oxford: Clarendon Press, 1997, p. 144.

[4] Goldratt, Eliyahu M., *Theory of Constraints*, Great Barrington, MA: North River Press, 1994.

CHAPTER 4

The Complete Single-Project Solution

This chapter describes the process of developing the single-project-management system to satisfy the system requirements identified in the previous chapters and further detailed herein. Although presented as a forward-moving process from requirements to design, the actual process, as is the case with nearly all designs, was iterative; that is, various design solutions were proposed and tested against the requirements until a suitable working system resulted.

4.1 From System Requirements to System Design

4.1.1 Requirements Matrix

Table 4.1 illustrates the requirements for an effective project-management system, following the method of Joseph Juran [1]. The table presents the requirements in a hierarchy, starting with the top-level necessary conditions for project performance. These include the three technical requirements for the project (scope, cost, and schedule) and the requirement for stakeholder satisfaction. Project stakeholders always include at least the project customer and the project team and may include many others (e.g., subcontractors, stockholders, regulators, neighbors, government, or other groups or institutions).

These are the requirements I use in defining CCPM. Many advocates of TOC and critical chain do not deal with the full scope of these requirements, for one or more of several reasons. Some are simply not aware of the full scope of requirements that the project system must meet, and they focus only on the part that Dr. Eliyahu Goldratt addresses. Others may not think they are necessary. I believe some organizations fail to achieve success with CCPM because they do not satisfy all of the necessary requirements for effective project delivery. A system that meets the requirements of Table 4.1 will satisfy all necessary conditions for many projects, although your project may have special requirements.

The second and third columns of the table illustrate the second- and third-level requirements derived from the top-level requirements. Requirements at the lower levels may vary for different types of projects; these are general requirements.

It is unlikely that you will generate an identical list of project requirements. This list includes elements of the PMBOK™, elements taken from my own experience, and elements specifically derived from the solution we are about to present. This feedback of the solution to the requirements is part of reality. Only by defining and critically assessing a proposed solution do we really understand the problem. In particular, before studying Goldratt and considering the basis of critical chain, I

would not have included accounting for common-cause variation among project requirements.

The table of project requirements is not, and can never be, complete. It is conjecture, a basis for criticism and improvement. For example, I am not satisfied that the requirements completely embrace profound knowledge, especially a knowledge of psychology. I suspect one could go further to ensure that the requirements capture

Table 4.1 Requirements for a Project System to Convert the Input of a Project Result Specification and Produce an Output of a Completed Project Result

Primary Requirement	*Secondary Requirement*	*Tertiary Requirements*
Define the project system	Project system goal	Define the project-system goal to complete projects that make money for the company now and in the future (for-profit companies).
	Project system boundary	Define the project-system boundary, starting with customer needs and ending with a satisfied customer.
	Account for understanding of variation	Account for common-cause variation in project processes.
		Provide a means to separate and deal with special-cause variation.
	Use the TOC to design the system	Identify the project constraint.
		Exploit the project-system constraint.
		Subordinate everything else.
	Include knowledge of psychology in the system design	Align project-system needs with individual psychological needs.
		Align individual rewards with project-system needs.
	Enable continuous improvement of the project system (a theory of knowledge)	Define and standardize processes.
		Measure process performance.
		Assess process performance.
		Improve processes.
Deliver the project result to the specification (scope)	Deliver all of the specified features	Satisfy all of the physical requirements for the specified features.
		Satisfy all of the functional requirements for the specified features.
		Satisfy all of the operational requirements for the specified features.
	Satisfy all of the feature quality requirements	Satisfy all of the feature quality requirements.
Deliver the project result on time (schedule)	Deliver the project result on time (schedule)	Complete the project on the quoted completion date.
		Complete intermediate milestones on the quoted completion dates.
Deliver the project result for the estimated cost	Maintain total project cost under budget	Complete the entire project for the quoted maximum cost.
		Do not spend more than specified maximums on subcategories of the total cost.
	Satisfy project cash flow requirements	Do not exceed project estimated cash-flow requirements.
Satisfy unique individual project stakeholders' needs (in addition to the above)	Satisfy project client requirements	Solicit and specify all requirements necessary to deliver a satisfactory final product.
		Provide evidence of meeting the project specifications.
		Provide information during the project to enable decisions that may affect the balance of the project.
		Respond to requests for changes.

Table 4.1 (continued)

Primary Requirement	*Secondary Requirement*	*Tertiary Requirements*
Satisfy unique individual project stakeholders needs	Satisfy Project team requirements	Provide the following: Clear scope definition; Clear responsibility and authority assignment; Project plan specifying who has to do what by when; Feedback to control performance to plan; Method to control interfaces with other team members; Method to raise and resolve issues during project performance; Change-control process.
	Satisfy subcontractors and supporting resources requirements	Provide the following: Clear scope definition; Clear responsibility and authority assignment; Project plan specifying who has to do what by when; Feedback to control performance to plan; Change-control process.

the principles of Six Sigma and Lean. I present it as a "good-enough" set of requirements to bind together CCPM and start us on a new path of project-system improvement to address the difficulties raised in Chapter 1.

4.1.2 Summary of Single-Project Critical Chain

Figure 4.1 illustrates the key features of the single-project critical-chain solution that satisfy the functional requirements for the project system. The illustrated features highlight the differences between CCPM and CPM. These essential features are:

1. Identifying the critical chain as the longest path through the project, considering both the task logic and the resource constraint;

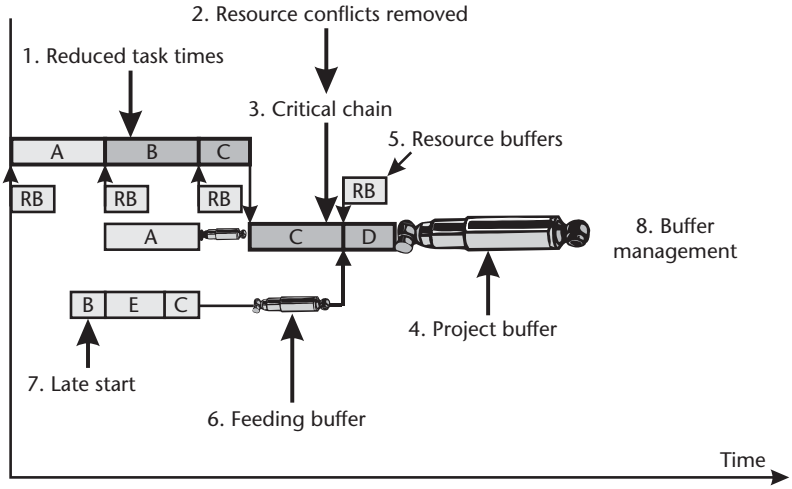


Figure 4.1 Key features of the critical-chain solution deliver performance to the project-system requirements.

2. Removing resource contention from the project plan before selecting the critical chain;
3. Exploiting the plan with mean (~50–50) task estimates, aggregating allowance for common-cause variation and bias into the buffers at the end of task chains (Figure 4.1 illustrates the buffer as a shock absorber);
4. Subordinating merging paths with feeding buffers (while continuing the elimination of resource conflicts);
5. Ensuring resource availability, especially for tasks along the critical chain (the method illustrated identifies resource buffers as one tool for this; I will describe others);
6. Using the project and feeding buffers as measures to control project performance.

The next section describes each of these features in greater detail.

Four essential behavior changes are required to use single-project CCPM effectively:

1. Management must encourage the use of mean task-duration estimates by not pressuring people to perform to the estimated durations.
2. Management must enable people to focus on one task at a time.
3. Resources must focus on one task at a time and pass on the results as soon as the task is complete.
4. Everyone must use the plan and the buffer reports to decide what to work on next.

That's it!

4.2 Developing the Critical-Chain Solution

The following sections describe the single-project critical-chain features in terms of the TOC focusing steps. I don't know if this is how Goldratt actually derived these features. Following Popper's [2] description of a theory of knowledge and the scientific method, how Goldratt defined these features (which Popper would have called "bold conjectures") does not matter. What does matter is that we subject the conjecture to critical discussion and test to see if the discussion supports the selection of critical chain over critical path.

4.2.1 Identifying the Project Constraint

The evident constraint of a project is the chain of tasks that takes the longest time to complete. The PMBOK™ Guide defines this as the critical path, "the sequence of schedule activities that determines the duration of the project. Generally, it is the longest path through the project." It then defines the CPM as

> A schedule network analysis technique used to determine the amount of scheduling flexibility (the amount of float) on various logical network paths in the project schedule network, and to determine the minimum total project duration. Early dates

> are calculated by means of a forward pass, using a specified start date. Late dates are calculated by means of a backward pass, starting from a specified completion date, which sometimes is the project early finish date calculated during the forward pass calculation.

There are several important, but unstated, considerations in these definitions. First, there is an assumption of a single (deterministic) task duration. Second, there is no information regarding the probability that one should attach to that duration. Is it a 50–50 duration, or is it a high-probability duration? The difference is a crucial.

The PMBOK™ Guide also describes probabilistic scheduling approaches, including PERT and Monte Carlo simulation. The PMBOK™ definitions of these approaches are too brief to be directly usable. My surveys indicate they are little used in practice.

In order to perform any task on a project, two things are necessary: the task input from a predecessor and the resource to perform the task. (The predecessor may simply be a start authorization for the first task in a chain of project tasks.) The definition of the critical path does not address the potential resource constraint. The resource need is implicit in the critical-path definition for many tasks because the task duration assumes a specific level of resource availability. The critical-path definition does not treat the constraint of resources across project tasks and does not allow the critical path to jump logic paths.

The basic definition of the critical chain is to simply identify the constraint of the project, or "The sequence of dependent events that prevents the project from completing in a shorter interval. Resource dependencies determine the critical chain as much as do task dependencies."

Defining the constraint of a project in terms of the schedule derives from the impact that schedule has on project cost and project scope. Independent variables that influence a project result include the demanded scope, the project-system definition, and the resources available to work on the project. The project system outputs are dependent variables (delivered scope, cost, and schedule). As schedule increases with fixed deliverable scope, cost usually increases. As scope increases with fixed cost (or resources), schedule tends to increase. As scope increases with fixed schedule, cost tends to increase. Therefore, it is appropriate to focus first on delivering the project on time.

Critical-path project planning contains a hidden assumption that an acceptable way to account for potential resource constraints on the project is first to identify the critical path and then perform resource leveling. Network specialists know that there is no optimum method for resource leveling. Some resource-leveling algorithms give very poor results. Certain resource loading approaches also contribute to poor results, usually blamed on the algorithm. For most networks, application of the resource-leveling algorithms lengthens the overall schedule. For this reason, few projects use the resource-leveling tools.

I have conducted an informal survey while delivering lectures to members of the PMI. These groups include many certified PMPs. Nearly all agree that getting the resources they need to work on their projects when they need them is difficult and often causes project delay. Yet, very few (<5%) indicate that they routinely resource-level their plans (i.e., account for the resource constraint within their own

project). When I ask why, those who respond most often state that resource leveling causes the plan length to exceed management demands.

Figure 4.2 illustrates a typical critical-path project schedule. The names represent unique resources. Evidently, we would fail to meet the schedule on the project because each resource can only do one task at a time.

Figure 4.3 moves tasks to eliminate the overlap of resource demands. In a manner similar to many computer algorithms for resource leveling, we first give the resource to the path with the least float, usually the initial critical path. Note that when we are done, all paths have float, so there is no critical path defined as the path with zero float. (Computer software packages treat this result differently. Some keep the initial critical-path definition. Some only define the last task as critical. I have no idea what they then do about the critical path as the project progresses and the critical path is supposed to change.)

More importantly, the initial critical path is not the constraint to completing the project. Since the resource constraint is often a significant project constraint, the TOC method of project planning always considers it. Thus, the critical chain includes the resource dependencies that define the overall longest path (constraint) of the project. The method resolves all resource constraints while determining the project critical chain. The project critical chain may have gaps between tasks. Figure 4.4 illustrates the critical chain for the example project.

If your organization does not have resource constraints (or has infinite resources), the critical chain will be the same initial task path as the critical path. This is an important fact in verifying the integrity of the critical-chain method; it contains the CPM as a special case, at least in regard to defining the critical chain. This is an important issue to validating the critical-chain method using the scientific method as the theory of knowledge.

An earlier PMBOK™ Guide definition of the critical path stated that it may change during the performance of the project. This can occur whenever project tasks experience common-cause variation, and it redefines the longest zero or negative float path to complete the project. Due to our knowledge of variation, this means that we should expect the apparent critical path to change frequently. Deming noted that one of the more serious mistakes managers can make is to treat common-cause variation as if it were special-cause variation. This PMBOK™ Guide definition of critical path and its implementation in many project-management systems institutionalize this mistake. This does not enable the project team to focus on the constraint to the project but causes them to make the error of chasing an ever-changing critical path. As Deming illustrated with the funnel experiment, this will always make project-system performance worse.

The critical chain does not change during project performance. This is partly a matter of definition, but mostly a result of the overall critical-chain plan-construction procedure and the subordination step described below.

4.2.2 Exploiting the Constraint

Having defined the critical chain as the constraint to performing the project faster, we now look to exploit the constraint. This means reducing both the planned time and the actual project performance time. CCPM exploits the critical chain using an understanding of variation. This is where Goldratt's unique focus on statistical

WBS	Task Name
1	Component 1
1.1	Task 1
1.2	Task 2
1.3	Task 3
2	Component 2
2.1	Task 4
2.2	Task 5
2.3	Task 6
3	Component 3
3.1	Task 7
3.2	Task 8
3.3	Task 9
4	Integration
4.1	Assemble
4.2	Test
4.3	Project Complete

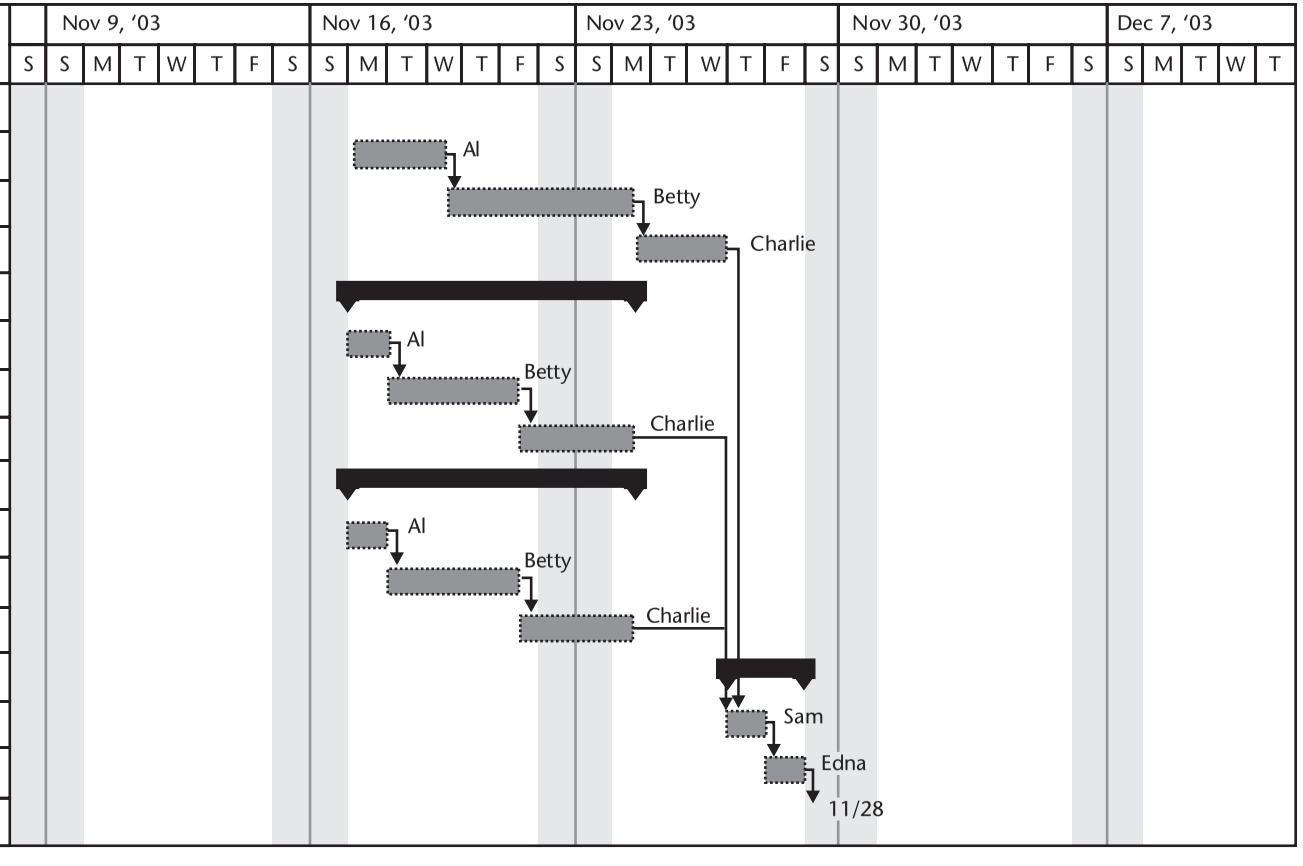


Figure 4.2 The critical path does not account for the resource constraint.

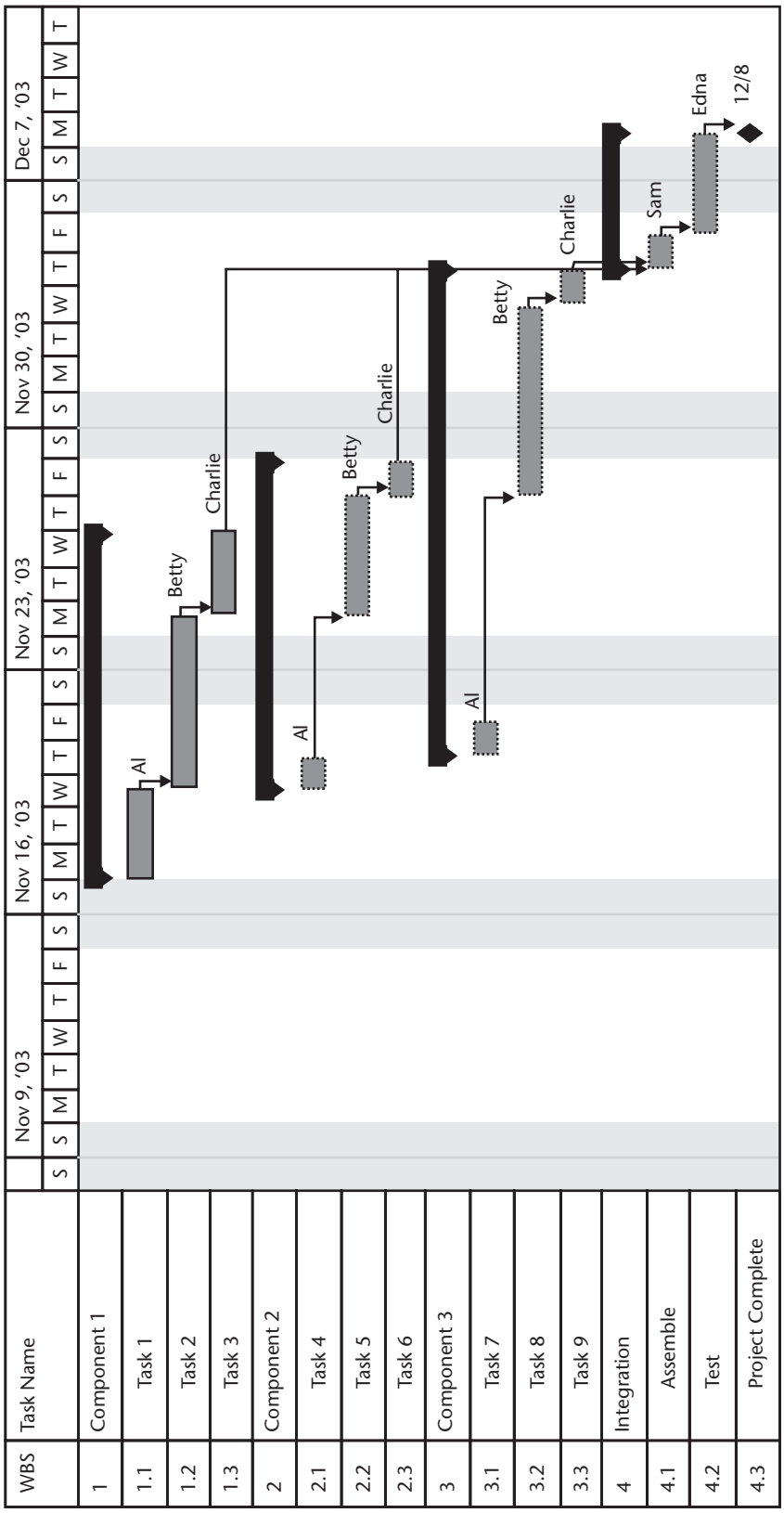


Figure 4.3 Removing resource conflicts usually creates gaps in the critical path.

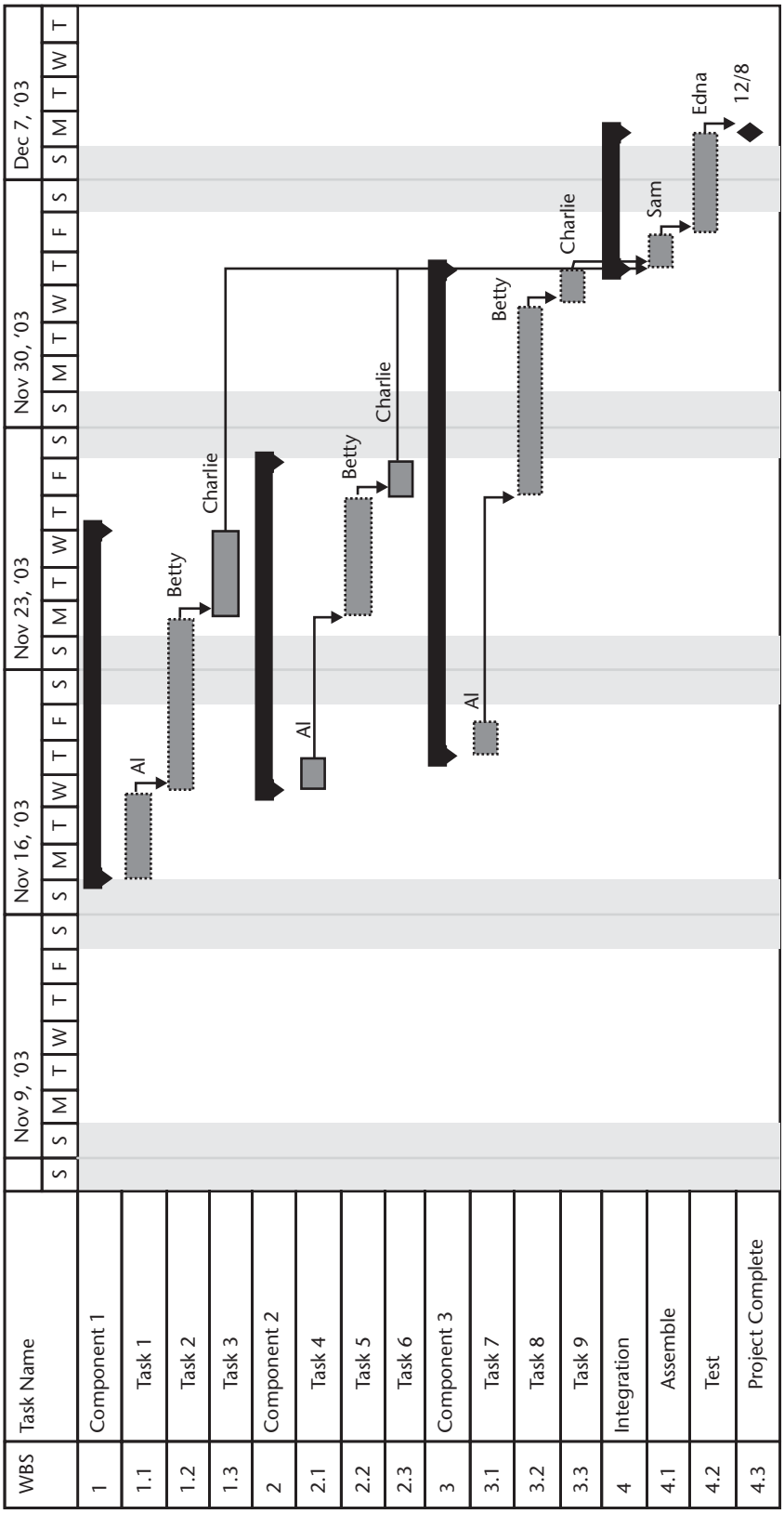


Figure 4.4 The critical chain includes both the resource and task logic constraints to completing the project on time or sooner.

fluctuations and dependent events leads to a significant departure from most current project systems. Goldratt's recognition of variation is not unique; but his solution applied to the project-management system is an innovation.

Deming notes that managers often make many systems worse by not understanding the fundamental difference between common-cause and special-cause variation. He also notes, "I should estimate that in my experience most troubles and most possibilities for improvement add up to propositions something like this:

94% belonging to the system (responsibility of management);
6% special."

Projects have common-cause variation in the performance time of tasks. Although the times necessary to perform individual project tasks may be independent of each other, project-task networks define task dependence. By the definition of the project logic, the successor task cannot start until the predecessor task is complete (for the most frequent, finish-to-start task connection.)

The TOC improvements for production take advantage of (exploit) the reality of statistical fluctuations and dependent events. Figure 4.5 illustrates a typical project task-performance time distribution. The solid curve (left ordinate) shows the probability of a given time on the abscissa. The dotted line shows the cumulative probability of completing the task in a time less than or equal to the time on the abscissa. Note the left skew of the distribution and the long tail to the right; this is typical of the common-cause variation for many project tasks.

Fluctuations in the actual performance of unique project tasks are likely to be much larger than fluctuations in the time it takes a production machine or person to repeatedly process a part. The project-task network clearly shows the many dependencies that exist in a project. Comparison of nearly any project to a

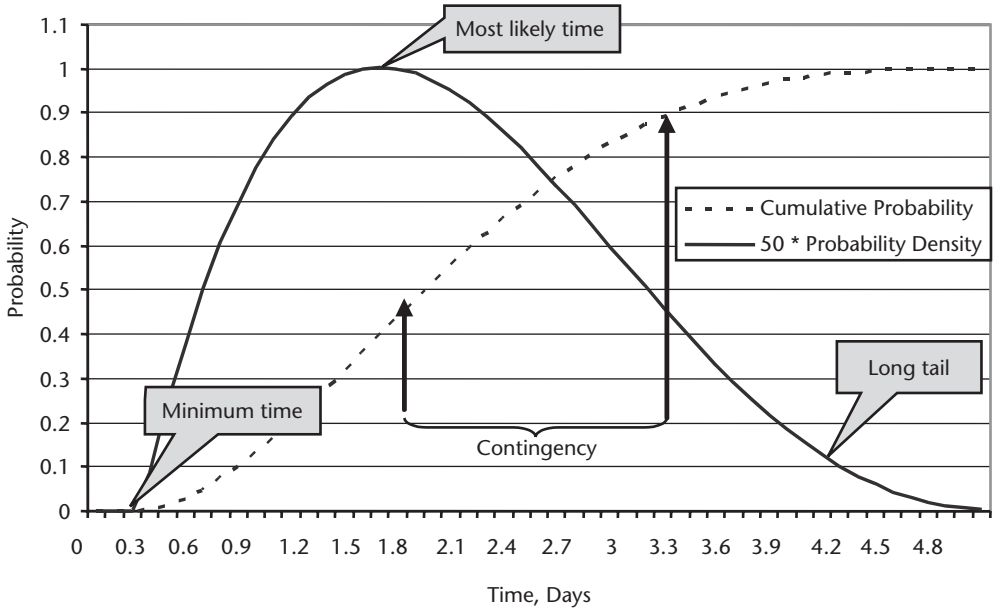


Figure 4.5 Typical project task-performance-time probability distribution illustrates key features: a minimum time, a most likely time, and a long tail to the right.

production line shows that there are more dependencies in even a modest-sized project. For these reasons, the logic that improved production should also improve project management.

This common-cause variation in task performance is not an exceptional event, such as discrete project-risk events. PERT attempted to estimate the impact of this common-cause variation using three task-duration estimates, but for a variety of reasons, it did not succeed. The PMBOK™ Guide and literature still make mention of PERT in this fashion, although it is little used today. PERT diagrams, as referred to in much of the project literature and in many project software packages, simply show the project-network logic independent of the time scale; they are not an application of the three time estimates. Some projects use methods such as simulation and Monte Carlo analysis to assess the impact of task duration and cost uncertainty. While these methods propose a way to estimate uncertainty, they do not pose an effective systematic method to manage it.

CCPM accounts for common-cause variation as an essential element of the project-management system. The process removes identifiable special causes of variation, including resource unavailability and common resource behavior patterns, including the student-syndrome and multitasking (described in Chapter 3). CCPM project managers use prioritized task lists, resource flags, or other approaches to identify and ensure availability of resources for tasks.

4.2.2.1 Exploiting Project-Task Estimates

CCPM seeks to use mean, or approximately 50% probable, individual task-time estimates. The CCPM project manager recognizes that actual individual task performance times include common-cause variation, and he or she does not criticize task performers for individual task-duration performance.

Most project managers attempt to account for individual task common-cause variation by adding contingency time into each estimate. They usually do not specify the existence or amount of this contingency time. People estimating task times for a project usually do so believing that the project manager wants "low-risk" task times, perhaps a probability of 80% to 95% probability of completion within the task-duration estimate or less. Figure 4.5 illustrates that this estimate is two or more times the 50% probable estimate. In most project environments, people feel good if they complete a task by the due date and feel bad if they overrun the due date. This reinforces their attempts to estimate high-probability completion times.

Walter A. Shewhart, mentor to W. Edwards Deming, said [3],

> It should be noted that the statistician does not attempt to make any verifiable prediction about one single estimate; instead, he states his prediction in terms of what is going to happen in a whole sequence of estimates made under conditions specified in the operational meaning of the estimate that he chose.

This view clarifies why attempts to deal with uncertainty for individual task estimates are fruitless.

I read and hear much about "improving the accuracy of estimates." I used to think that this was a good thing to do, that if we would apply a more disciplined process, we could do a better job of estimating the time or cost of a project.

Actually, I know that to be true, but understanding variation changed my understanding of what that means. Most people, when they address "improving the accuracy of estimates," have in mind improving the accuracy of each point estimate that sums to the total cost or duration. Shewhart clarifies that *you cannot do this*. Indeed, I have come to realize that the probability of all point estimates is exactly the same: zero. You only have a finite probability when you state an interval that a single result might fall within. Thus, accuracy, as most people mean it, does not exist.

Consider defining the accuracy of a gun. If you shoot one shot at a target, you have no idea of how accurate the gun may be. Common-cause variation may have put that one shot right on target, or several inches off, or more. The only way to determine the accuracy of the gun is to shoot a number of shots, measure the spread of the result, and compare the center of the spread to the center of the target. Of course, you aren't really measuring the accuracy of the gun in that case, either. You are measuring the accuracy of the gun, the shooter, the cartridge, and the environment. Changing any of them will change the apparent accuracy of the gun. For example, letting my son shoot instead of me will make for a much more accurate gun. Shooting at 25 yards versus 100 yards will make for a much more accurate result. And so on.

Understanding what accuracy really means (the variation in the result relative to the specified mean result) clarifies that there are two ways to improve the accuracy of single-point estimates. You can better define the estimate assumptions, thus narrowing the necessary band (standard deviation), or you can improve the process. For example, specifying the gun accuracy at a specific range with a specific cartridge and a specific shooter will yield a smaller range in the variation (improve the uncertainty of the accuracy). Also, doing things such as clamping the gun or shooting indoors where there is no wind will actually change the process and reduce the variation. The counterpart to the first approach would be an improved estimating process (e.g., using an estimation database and repeatable process versus ad hoc estimates). The counterpart to the second approach would be improving the actual work process (e.g., using written procedures or new tools).

Some experienced project managers state that people tend to give optimistic estimates. They base this contention on remembering the instances in which projects had difficulty meeting the delivery date. Generalizing this observation does not hold up under examination for several reasons.

First, extensive psychological research demonstrates that people tend to seek pleasure and avoid pain. In most project environments, people get pleasure and avoid pain by completing tasks on the due date. Hardly anyone wants to be known as the person who can be counted on to deliver late. It is not reasonable to expect people to solicit pain by systematically giving "optimistic" estimates.

Second, people remember selectively. They easily remember worst-case outcomes (pain) but not necessarily all of the times things went to their advantage. Don't most people feel that they always pick the slowest line in a bank or supermarket? Do you really believe that this is true? People also tend to forget predecessors leading to the outcome. This mental feat has two interesting effects:

1. The project manager selectively remembers the instances where task-duration estimates were exceeded and therefore wants to add contingency of his own.

2. Task performers tend to add time to their next estimate.

Third, if underestimating task durations were the predominant fact, nearly all project tasks would be late. Assuming that most of the potential positive variation in task times is returned to the project (evidence suggests otherwise), the merging of task paths ensures a very low probability of success if individual estimates are less than 50% probable. (Real project behavior is, of course, confounded by control actions taken during project performance. These control actions may help or hinder overall completion time performance.)

While many projects do fail to meet schedule, our observations indicate that a substantial portion (e.g., about one third of information-technology projects) do achieve the scheduled project end date. Almost all projects to create bid proposals complete on time. Nearly all major meetings come off as planned with few problems. The Olympics has not yet been delayed due to late project completion. (Greece caused anxious moments in 2004, but was ready nonetheless.)

Milestone performance in a very large project I examined demonstrated that the task-performance data conformed very closely with Goldratt's prediction that about 80% of the task milestones are achieved exactly on the scheduled date, with only one or two sooner and the rest later, including a few significantly later. This project consisted of about thirty large subprojects, some of which contain yet smaller subprojects.

My experience shows project plans from a variety of organizations (numbering in the hundreds) either fail to specify what probability and confidence of estimate are expected for task-duration estimates, or they fail to provide a quantitative basis for the estimate, or both. The PMBOK™ Guide admonishes project managers to provide these estimates but provides little guidance on what to do with them. Construction projects are somewhat of an exception, having access to extensive quantitative data. For example, the *National Construction Estimator* [4] uses an extensive database. The *Construction Estimator* lists many potential contributors to common-cause uncertainty in the estimates. The guide states that many of these uncertainty items range within several tens of percentage points of the cost estimate. Therefore, in many cases, they have the same potential impact on schedule.

4.2.2.2 Exploiting Statistical Laws Governing Common-Cause Variation

CCPM exploits the statistical law of aggregation by protecting the project from common-cause uncertainty of the individual tasks in a task path with buffers at the end of the path. Buffers appear as tasks in the project plan but have no work assigned to them.

Note that in the statistical terminology, variance is the square of the standard deviation, usually represented by s^2. For a given statistical distribution, it requires a given number of standard deviations to provide a cumulative probability to that point. For example, with a normal distribution plus or minus one standard deviation includes 67% of the data, or a cumulative probability that 67% of the time; a result will fall within one standard deviation of the mean.

The statistical method to combine variances means that you can protect a chain of tasks to the same level of probability with much less total contingency time than you can protect each individual task. Aggregation of the contingency times dramatically reduces the overall estimated time for a chain of tasks.

Figure 4.6 illustrates, for a simple case, how the law of aggregation leads to a shorter schedule. For the case shown, we assume that each of the four tasks has a 50% probable time of one week and a 90% probable time of two weeks. Therefore, the chain of four tasks has a planned time of eight weeks. Based on the student syndrome and date-driven behavior, we would consider delivery unlikely before eight weeks and most likely later.

A second factor that comes into play in aggregating tasks is the central-limit theorem, described in Section 3.4. Many project tasks have a skewed probability distribution; that is, they have an absolute minimum time and a long tail to the right, meaning that they can take much longer than the average time. These left-skewed distributions also generally have a mean that exceeds the most frequent, or median, time. A project chain of tasks is therefore more likely to have a symmetrical distribution and a variance that is much smaller than the algebraic sum of the individual task distributions. This is true whether you know the real distributions or not.

The critical-chain plan uses the 50% probable times to create a critical chain of the four tasks that is four weeks long. The project buffer, the square root of the sum of the squares of the differences between the 50% time and the 90% time (each one week), is two. Therefore, the total project plan is only six weeks. Considering date-driven behavior, there is some chance the critical-chain project will complete in four weeks.

4.2.2.3 Exploiting Resource Availability

One of the leading, alleged causes of late projects is that resources are not available, or are not available in sufficient quantity, when needed. CCPM requires a mechanism to prevent the critical-chain tasks from starting late or taking longer due to the resource. Goldratt proposed a resource buffer to provide information to the critical-chain resources about when they will be needed. I rely on a more traditional approach, greatly facilitated by modern PC-based project software. The approach I recommend is to filter the project file for each type of resource and to give each a reasonable time window (perhaps three to four weeks) so that each resource or resource manager can see what tasks are upcoming. Doing this with the working schedule provides a continuous best estimate of when the task is likely to start. I avoid printing columns of dates on these reports, as dates rapidly lead people back into expecting deterministic dates. Instead, I let the calendar illustrate rough date intervals. Figure 4.7 illustrates an example.

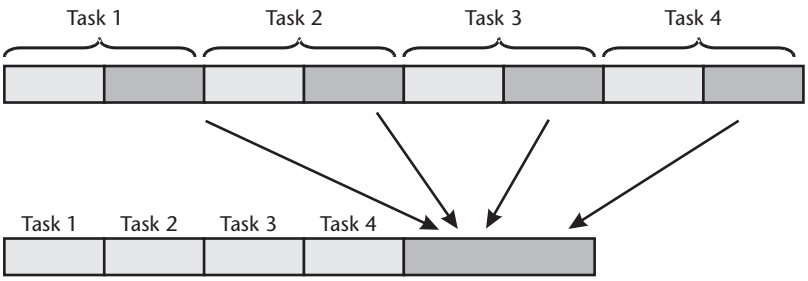


Figure 4.6 Aggregating uncertainty reduces planned project lead-time while maintaining the level of due-date protection.

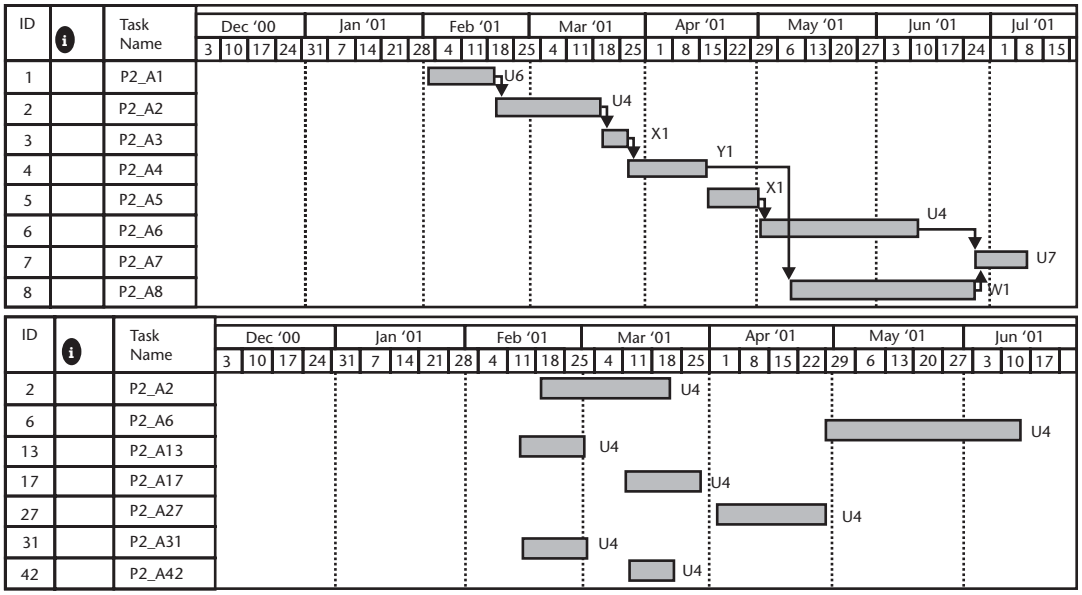


Figure 4.7 A filtered view of upcoming tasks by resource provides notification of roughly when tasks will be ready to work: (a) the upper figure presents only the first eight tasks in the project; (b) the lower view depicts all the tasks using resource U4.

Goldratt's resource buffer is different from the project buffer and feeding buffers described in the next section in that it does not occupy time in the project network. It is an information tool to alert the project manager and performing resources of the impending necessity to work on a critical-chain task. Note that inherent in the critical-chain idea is the notion that you cannot deterministically schedule resources. Since each task performance will vary, any forward-deterministic schedule is an uncertain estimate. You establish the lead-time necessary for each resource on the critical chain of the project and use the project measurement-and-control process to alert the resource as the time of actual task performance approaches. You may use multiple notifications (e.g., at one month, one week, and two days). You may use different times for different resources or use a standard time. You may choose to use alternative methods for subcontracted resources, such as contract rewards or penalties for delivering to a specified lead-time or duration.

4.2.3 Subordinating Merging Paths

Most projects have multiple task paths. All task paths must merge into the critical path by the end of the project, if for no other reason than to achieve a milestone that identifies project completion. Usually, the path merges tend to concentrate near the end of the project. One reason for this is that "assembly" or "test" operations tend to occur near the end of the project, requiring many elements to come together. The following demonstrates how this becomes a primary cause of the well-known project "truth" that many projects complete 90% in the first year and the last 10% in the second year. Figure 4.8 illustrates the filtering effect of merging paths. The successor task cannot start until the latest of the predecessor tasks is complete.

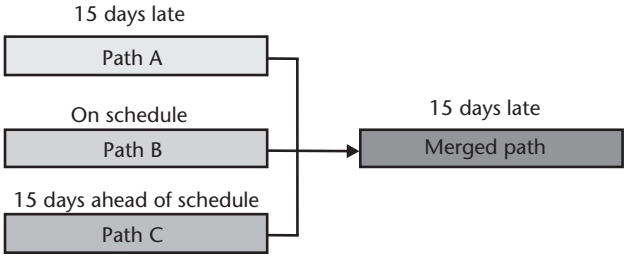


Figure 4.8 Merging paths cause critical-chain delay if any of the feeding chains are delayed.

Task-path merging creates a filter that eliminates positive fluctuations and passes on the longest delay. The reason is that merging task paths means that all of the feeding paths (outputs of the predecessor tasks) are required to start the successor task. Therefore, the successor task cannot start until the latest of the merging tasks completes. Consider a task on the project critical path that requires three separate inputs in order to start. This is very common in assembly operations and in many project results, such as a major show or meeting event, where everything has to be ready on opening day. Usually, there are many more than three. However, even with three, if each has a 50% chance of being done in the estimated time, the probability that at least one is late is over 88%! Even if each individual task had a 90% probability of completion, the probability that at least one will be late is still 30%, or nearly one in three.

CCPM protects the critical chain from potential delays by subordinating critical-chain feeding paths, placing an aggregated feeding buffer on each path that feeds the critical chain. Figure 4.9 illustrates the placement of the feeding buffers. This includes paths that merge with the critical chain at the end of the project. The feeding buffer provides a measurement-and-control mechanism to protect the critical chain. Figure 4.9 illustrates how the buffers absorb the late paths.

This innovation immunizes (to an extent) the critical chain from potential delays in the feeding paths. It also provides a means to measure the feeding paths, while keeping focus on the critical chain.

Many experienced project managers have confused the feeding buffer with project float, or slack. They are very different. You size feeding buffers based on the variation in the chain of activities that precede the feeding buffer. Thus, its size depends on the variation of the task. Inserting a properly sized feeding buffer may cause a gap in the critical chain.

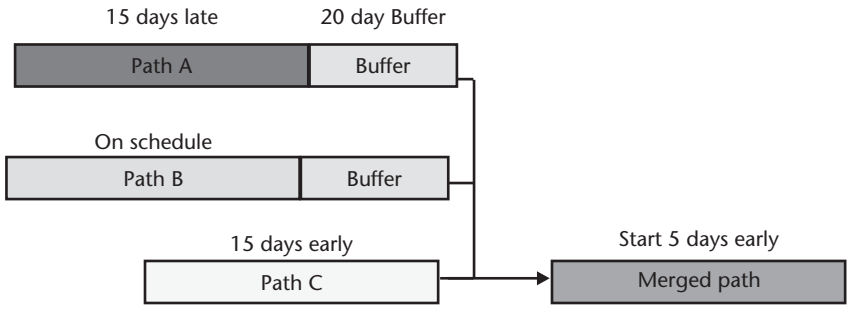


Figure 4.9 Feeding buffers absorb fluctuations in critical-chain feeding paths.

Float, or slack, is a result of calculating the network using deterministic single-point task durations. It has nothing to do with the variation of task duration. A chain of tasks that is nearly as long as the critical path gets near-zero float, or slack, and probably requires the most relative to other chains. The idea that float, or slack, can help protect the network from merging paths is fundamentally flawed.

4.2.4 Task Performance

4.2.4.1 Elevating Date-Driven Performance

The primary local optimum of significance in project management is the estimate of each individual task time. If management judges the performer of each task based on his or her completing a task on the estimated milestone date (local optima), what does this do to the overall project completion time (system optima)?

Critical-chain project plans only provide dates for the start of task chains and the end of the project buffer. For the rest of the project, the plan provides approximate start times and estimated task remaining duration. Critical-chain project managers do not criticize performers that overrun estimated task durations as long as the resource (a) started the task as soon as he or she had the input, (b) worked 100% on the task (no multitasking), and (c) passed on the task output as soon as it was completed. They expect 50% of the tasks to overrun.

Analysis of almost any project's results reveals that people report very few tasks as completed early. If you had 50% estimates, you should expect that people complete and report 50% of the tasks early. If you have 99% estimates, 99% of the tasks should be reported completed early. Usually, people report most tasks as completed on the task finish date, and they report some of the tasks as late. Why don't projects see much positive variation?

The reason is that, most of the time, performing resources fail to pass on most of the positive variations. *Critical Chain* describes several effects that lead to performers systematically overrunning these times, although they initially had extensive contingency time. Jack R. Meredith and Samuel J. Mantel [6] state, "The operation of Parkinson's Law presents a clear and present danger. The work done on project elements is almost certain to 'expand to fill the additional time.'" In Goldratt's words, the safety time is wasted.

In the business and government cultures I have witnessed, there is little or no reward for completing individual tasks early, and there is some type of punishment or negative feedback (even if self-inflicted) for being late or having quality problems. In many project environments, there is a significant *disincentive* to reporting a task as completed early. Work performed on "time and material" contracts results in less revenue if the work is completed and turned in early. Many companies budget work performed by internal functional organizations as if it were time and material contract work. If the functional organization completes the work in less time than estimated, it cannot continue to charge to the project and must find alternative work for the resources. If individuals complete tasks early, they get more to do. These cultures drive local optima, which means delivery on the milestone date, but not before.

There are many ways to justify keeping the potentially early result. Managers can put its review or completion at "low priority" because it is not due yet, or the

resource can "polish the apple." The result is the same: people waste contingency time originally included in individual task-time estimates.

Chapter 3 described the student syndrome, illustratd by Figure 3.10. Many people have a tendency to wait until tasks get really urgent before they work on them. This is especially true for busy people in high demand; that is, all of the most important people the project manager is counting on to get the critical-path work done on time. If people believe they have some extra time in their estimates, they are often willing to accept other "higher priority" work at the beginning of the scheduled task duration. This tends to waste their contingency time, forcing them to perform most of the work in the later portion of the scheduled task time.

The combination of aggressive task duration initial estimates (nominally one half of previous estimated, elimination of task due dates, and frequent update of task estimated remaining duration conspire to eliminate date-driven behavior.

Critcal chain usses the metaphor of a relay-racer to describe expected task performance. Once you have the input to your task (the baton), you work on it full speed to pass the baton (your result) on to the next task in the project

4.2.4.2 Elevate Task Performance by Eliminating Multitasking

Multitasking is the performance of multiple project tasks at the same time. Some people refer to it as the "fractional head count." Humans are not too good at rubbing their tummies and patting their heads at the same time. People actually multitask by dividing out time between the multiple tasks. People might do this during the course of the day by working on one project in the morning and one in the afternoon.

Most people think of multitasking as a good way to improve efficiency. It ensures everyone is busy all of the time. Often I have to wait for inputs or for someone to call back before I can get on with a task. Multitasking makes good use of this time.

Goldratt demonstrates in *The Goal* how focus on local efficiency can damage the overall performance of a system. He uses the example of robots, which are operated all of the time in order to show high efficiency. In the case of production, this produces excess inventory and may plug the constraint with work not necessary for current orders, increasing operating expenses and delivery times, with no positive benefit to the company as a whole.

Multitasking on project tasks has a much worse impact. Consider a person who has to do three one-week tasks for three different projects (see Figure 4.10). If that person were permitted to work exclusively on each one, the first project would have its result in one week, the second project at the end of the second week, and the third project at the end of the third week. If the task performer multitasks, spending for example one third of his or her time each day on each project, none of the projects gets its output until the end of the third week. All three tasks have a three-week duration, potentially extending the overall duration of each project.

If multitasking is a normal way of doing business in a company, three weeks becomes the normal task duration. Performance data supports this inflated task duration. If these are critical-chain tasks, the practice directly extends project durations. Most companies admit to encouraging extensive multitasking.

CCPM seeks to eliminate this type of multitasking by eliciting 100% focus on the project task at hand by all resources supporting the project. Thus, eliminating

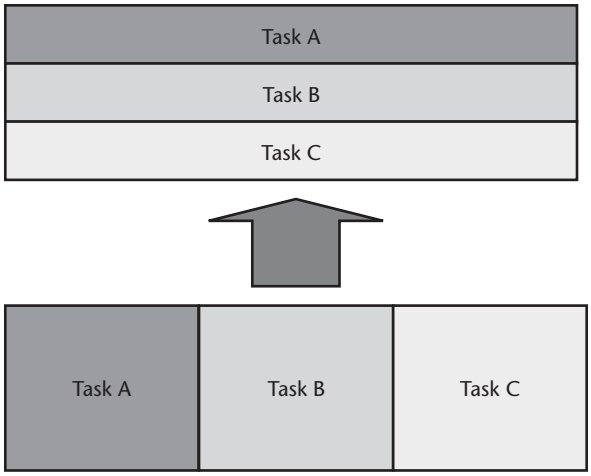


Figure 4.10 Multi-tasking extends project task duration

"fractional head counts" is a primary consideration in planning a critical-chain project. CCPM also requires providing resources information to determine which task I work on next.

I am often asked, "Isn't it a manager's job to multitask?" or "What if I am held up on one project task?" My answer is to clarify that there can be good multitasking. *Bad multitasking is multitasking that extends the duration of a project task*. As long as you position yourself and your project work to avoid bad multitasking, you are contributing your best to the project team.

4.2.5 Early Start versus Late Finish

Extensive studies have evaluated the desirability of using early-start schedules or late-finish schedules. Project managers believe early-start schedules reduce project risk by getting things done early, and late-finish schedules do the following:

- Reduce the impact of changes on work already performed;
- Delay the project cash outlay;
- Give the project a chance to focus by starting with fewer simultaneous task chains, allowing the project team and processes to come up to speed.

Some project-management guidance recommends that project managers use an early-start schedule. Many schedule computer programs use the early-start schedule as the default. Early start means permitting all of the non-critical-path tasks to start earlier than is necessary to meet the schedule date. People working on those tasks know that there is slack in their task. How do you think this influences the urgency they feel in working on their tasks? Does it encourage or discourage the student syndrome?

CCPM can default to late start for project tasks. The feeding buffers provide an explicitly sized buffer to protect the overall project from late completions in the feeding paths. This maximizes the advantages to the project, while ensuring project-schedule protection.

Some projects require early starting of selected tasks for risk mitigation. For example, projects that renovate or repair complex systems (e.g., ships) usually do not have an explicit idea of the work to be performed until they perform some type of diagnosis. It makes sense to do these diagnosis tasks as early in the project as possible in case they discover the need for lengthy repair work or work requiring long lead-time materials. You should force the scheduling of these tasks early in the project.

I am often asked, "What does it hurt to start early if I have the resource?" I answer by agreeing that once you understand this theory, if it does not hurt anything, by all means do it. TOC requires that people use their knowledge.

4.3 Exploiting the Plan Using Buffer Management

Measures drive actions that move you toward the goal. In *The Haystack Syndrome,* Goldratt notes [7],

> The first thing that must be clearly defined is the overall purpose of the organization—or, as I prefer to call it, the organization's goal. The second thing is measurements. Not just any measurements, but measurements that will enable us to judge the impact of a local decision on the global goal.

Figure 4.11 illustrates the "cybernetic view" of measures used by Juran. The sensor makes the measure in block 2. An umpire (block 4) compares the output of the process as reported by the sensor to the goal for the process. The umpire makes a decision to cause an action, modifying the process to change output and minimize the gap. This is how all control systems work. This is the intent of project-measurement systems, where the goal includes the technical requirements, cost, and schedule for the project.

In *The Haystack Syndrome* [7], Goldratt defines *data* as, "every string of characters that describes something, anything, about our reality." He defines *information* as "the answer to the question asked." Goldratt suggests that the information system should incorporate the decision.

The improved measurement system for CCPM follows the practice established by Goldratt for production operations. It uses buffers (i.e., time) to measure

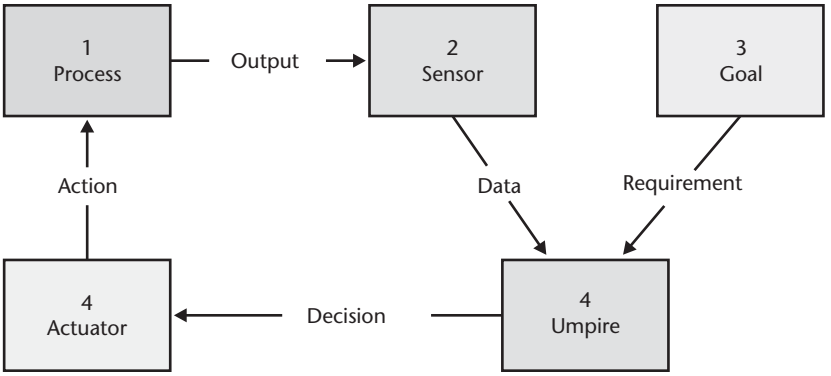


Figure 4.11 Dr. Joshep Juran depicts measurement as part of a control process.

task-chain performance. You size the buffers based on the length of the task chain they protect. Buffer sizing uses the uncertainty in the duration of the critical-chain tasks to size the project buffer. Likewise, uncertainty in the duration of the feeding-chain tasks determines the size of each critical-chain feeding buffer. CCPM sets explicit action levels for decisions. The decision levels are in terms of the buffer size, and the percent of the critical chain that has been completed. See Section 6.4.3 for a discussion of buffer trigger points. The buffer is supended into three regions:

1. In the green region, take no action.
2. In the yellow region, assess the problem and plan for action.
3. In the red region, initiate action.

These measures apply to both the project buffer and the critical-chain feeding buffers. Figure 4.12 shows an example of using the buffers.

Project teams monitor the project buffer and each critical-chain feeding buffer at the appropriate time intervals for the project, often daily but at least weekly. For this tool to be fully useful, the buffer monitoring time must be the lesser of at least as frequent as one third of the total buffer time or as frequent as the shortest duration lasts in your project plan. If the buffers are negative (i.e., the latest task on the chain is early relative to the schedule date) or less than one third of the total buffer late (e.g., less than 10 days if the total buffer is 30 days), you do not need to take action. If extended durations penetrate the buffer to the yellow action limit in the project team should plan actions for that chain to accelerate the current or future tasks and recover the buffer. If the task performance penetrates the buffer beyond the red threshold, the project team should take the planned action. Through this mechanism, buffer management provides a unique anticipatory project-management tool with clear decision criteria.

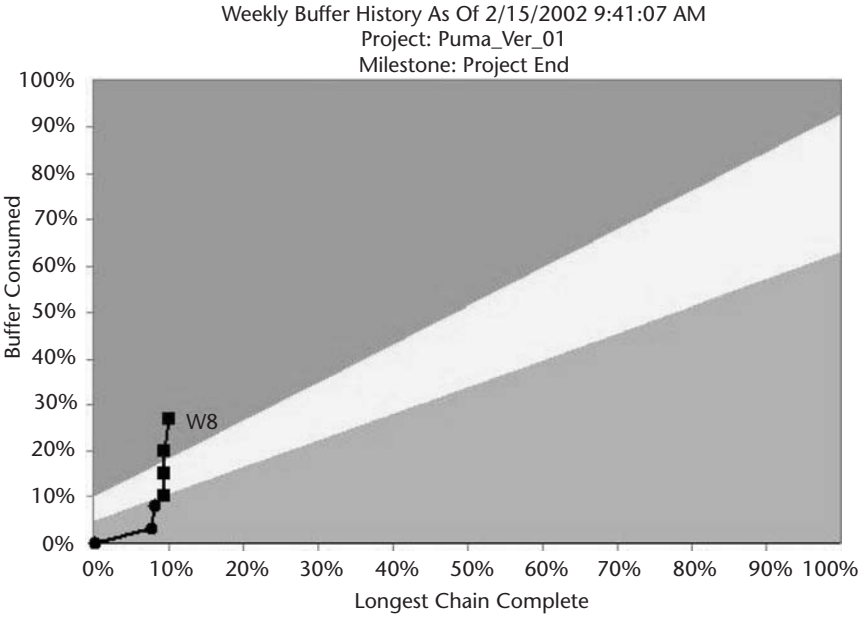


Figure 4.12 A fever chart plots trends of buffer-penetration versus dynamic-action thresholds.

Project managers update the buffers as often as they need to by simply asking each of the performing tasks how many days they estimate until the completion of their task, or the *remaining duration*. They do this without pressure or comment on the estimate. They expect these estimates to vary from day to day and for some of the tasks to exceed the original duration estimates. As long as the resources are working on the tasks within the CCPM task-performance paradigm, managers evaluate them positively, regardless of the actual duration.

An enhancement in the use of the buffer for long critical chains is to plot trends for buffer utilization. The buffer measure then becomes, in essence, a control chart and can use similar rules; that is, any penetration of the red zone requires action. Four points trending successively in one direction require action. Most critical-chain software has extended this idea to use a fever chart (Figure 4.12), where the buffer action thresholds may vary over the duration of the project. The idea is that you should not use up the project buffer too early in the project. Trending buffer data preserves the time history of the data and shows the trend of buffer consumption vs. project time. This information helps improve control of the schedule.

Updating the buffers requires that you maintain project status versus your plan in terms of the tasks completed, and estimated remaining duration for incomplete tasks. This is also a useful direct measure of project performance.

4.4 Features (More or Less) from PMBOK™

The unique features of CCPM do not comprise a sufficient system to satisfy the project-system requirements identified at the beginning of this chapter. The PMBOK™ Guide seems to provide all of the necessary additional features to meet the complete system requirements. Following Juran, I prepared a feature-and-requirement correlation matrix to examine how the CCPM features and selected features and processes from the PMBOK™ Guide combine to provide the complete set of identified project-system requirements. The table is too large for publication in book format. It helped to identify the following set of PMBOK™ Guide features as the primary ones necessary to deliver to the requirements given in Table 4.1.

Such a correlation table also leads to clarification of the requirements that pertain to each feature and, therefore, supports developing the feature.

The features described in the following sections are (mostly) contained in and explained in the PMBOK™ Guide and are necessary to satisfy the requirements.

4.4.1 Project Charter

The project charter authorizes the initial project team to prepare the project work plan. It identifies the overall project deliverable, project stakeholders, overall project responsibilities and other parameters necessary to create an effective project work plan.

4.4.2 Project Work Plan

The project work plan identifies the scope, budget, schedule, responsibilities, and resource requirements for the project. It may also specify other project requirements

and plans to achieve them, such as quality, safety, and regulatory plans. It must contain or reference the operational procedures for the project. Key elements of the project work plan are described below.

4.4.2.1 Work Breakdown Structure

The WBS is the framework to define project scope. It defines project scope hierarchically, from the complete project level to the work-package level. Work packages complete the hierarchy by specifying the project tasks necessary to deliver the scope.

4.4.2.2 Responsibility Assignment

Responsibility assignment designates individuals responsible to accomplish deliverables on the WBS. Responsibility assignment must occur at the work-package level and may be assigned at higher levels. Responsibility assignment normally confers the *authority* to perform the work and *accountability* for delivering the scope to the budget and schedule for the project deliverable. Responsibility assignment generally is not the same as the resources assigned to perform the task. Responsibility assignment must be made to a named individual; while resources assigned to perform tasks can be generic (e.g., pipe fitter or programmer). You should also assign unambiguous responsibility to a task manager for each task. The task manager is accountable for statusing the schedule for that task. The task manager may be a resource working on the task or a resource or work-package manager.

4.4.2.3 Milestone Sequencing

Milestone sequencing is a tool to go from the hierarchically formatted WBS to a logical project plan. It provides the major sequence of project tasks for use by work-package managers to link the inputs and outputs of their work packages. (This element is not described in the PMBOK™ Guide but is covered in the next chapter.)

4.4.2.4 Work Packages

Work packages define the plan to produce project deliverables at the lowest level. Work packages contain the scope definition for the deliverable of the work package and the plan to produce the deliverable. This plan includes defining the project tasks, the logic for the tasks, and the linkage of the work-package tasks to other elements of the work plan, usually to milestones on the milestone sequence chart. Work packages may link to tasks in other work packages as well, but this linkage usually cannot occur on the first draft as all work packages are planned simultaneously. Work packages also identify the estimated task duration and resource requirements, as well as the assumptions necessary to support these estimates.

4.4.2.5 Project Network

The project network logically connects all of the tasks necessary to complete the project. The project tasks must identify the resources necessary to perform the tasks

within the estimated task duration. The network includes all of the tasks from all of the work packages and identifies the critical chain, project buffer, and critical-chain feeding buffers. It provides start dates for each chain of tasks and the completion date for the entire project. It is the basis for subsequent performance measurement and control.

4.4.3 Project Measurement and Control Process

CCPM defines an improved schedule-measurement-and-control process. Most projects also require a technical-quality-control process, and many projects also require a cost-control process.

The correlation matrix also identifies a need for processes to ensure project result quality and provide mechanisms for continuous improvement. This scope of this text does not address the process to ensure project quality results. Lewis Ireland [8] provides an overview of a satisfactory process to meet these requirements.

4.4.4 Project Change Control

The project-measurement-and-control process will, from time to time, trigger the need for action to complete the project successfully. In addition, unfulfilled assumptions made at the start of the project, "as-found" conditions that differ from initial assumptions, or changes in the client's demands may require changes in the remainder of the project. Project change control defines a process to incorporate and communicate these changes to the entire project team.

4.4.5 Project-Risk Management

Project-risk management handles potential causes of special-cause variation. Since the PMBOK™ Guide does not differentiate between common-cause variation and special-cause variation, you will find it addresses both under the realm of project-risk management. CCPM addresses common-cause variation directly in the project system and, thus, confines project-risk management to special-cause variation. See Chapter 10.

4.5 Summary

This chapter has developed the project system requirements and described the critical-chain features and key supporting PMBOK™ system features designed to satisfy those requirements. Key system features are as follows:

- The critical chain identifies the project constraint.
- Exploiting the critical chain utilizes uncertainty management in the form of reduced task-plan durations and a project buffer.
- Exploiting the critical chain requires providing resources the answer to the question, "which task do I work on next?" You can do this with prioritized task lists, filtered resource views, or resource flags.

- You should subordinate feeding chains and resource efficiency to the critical chain with critical-chain feeding buffers.
- Critical-chain projects rely primarily on buffer management for project control.
- Additional features from the PMBOK™ Guide are necessary to complete an effective project-management system.

These high-level system features are necessary and sufficient to satisfy the requirements.

References

[1] Juran, Joseph J., *Juran on Planning for Quality,* New York: The Free Press, 1988.

[2] Popper, Karl R., *Objective Knowledge, An Evolutionary Approach*, Oxford: Clarendon Press, 1972.

[3] Shewhart, Walter A., *Statistical Method from the Viewpoint of Quality Control,* New York: Dover Publications, 1986 (originally published in 1939).

[4] Kiley, Martin D., *1997 National Construction Estimator*, Carlsbad, CA, Craftsman Book Company, 1996.

[5] Moore, David S., and George P. McCabe, *Introduction to the Practice of Statistics,* New York: W. H. Freeman & Co., 1993, p. 398.

[6] Meredith, Jack R., and Samuel J. Mantel, *Project Management, A Managerial Approach,* New York: John Wiley and Sons, Inc., 1995.

[7] Goldratt, Eliyahu M., *The Haystack Syndrome*, Croton-on Hudson, NY: North River Press, 1990.

[8] Ireland, Lewis R., *Quality Management for Projects and Programs*, Upper Darby, PA: PMI, 1991.

CHAPTER 5

Starting a New Project

5.1 Project-Initiation Process

The project-initiation process ensures meeting all of the conditions necessary for project success. It starts with a clear understanding and agreement among all of the project stakeholders on the expected project results and ends with a clear understanding of who is responsible and accountable for doing what, by when, to achieve the result.

Figure 5.1 illustrates an overall process to successfully initiate a project. It starts with the project charter, an often overlooked but necessary part of any project. It ends with a project work plan that is sufficient to start work on the project.

The PMBOK™ Guide [1] separates the project-initiation process from the project-planning process. This chapter considers both of them as part of project initiation. With this definition, the outputs of project initiation include the project charter, project manager assignment, project constraints, project assumptions, and other elements of the complete project plan, including the schedule and budget.

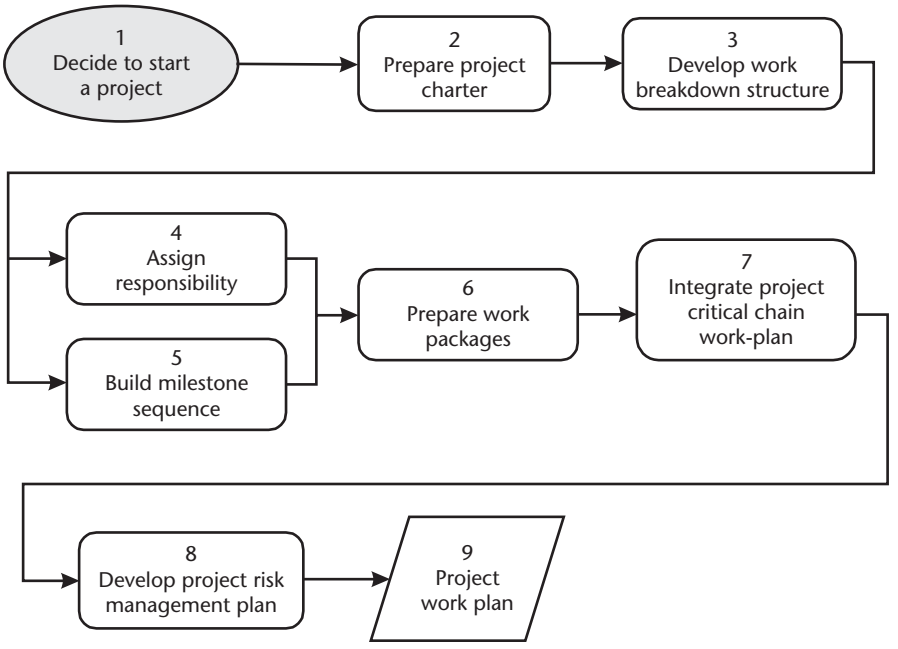


Figure 5.1 The project-initiation process.

5.2 The Project Charter

A project charter is a brief written statement to enable the assembly of an effective team to plan the project. This definition goes well beyond the charter described in the PMBOK™ Guide, including summarizing the "five Ws and an H" (i.e., what, when, who, why, where, and how) to plan the project. It should normally include all of the elements described by CH2MHILL [2] as essential:

- Vision;
- Purpose;
- Membership;
- Mission;
- Organizational linkage;
- Boundaries;
- Team and individual responsibilities;
- Measures of success;
- Operating guidelines.

The primary distinction between the project charter and the project work plan is that the charter authorizes developing the work plan.

5.3 Stakeholder Endorsement

The PMBOK™ Guide defines project stakeholders as

> individuals and organizations that are actively involved in the project, or whose interests may be positively or negatively affected as a result of project execution or project completion; they may also exert influence over the project and its results. (p.16)

For most projects, this list includes a variety of people and organizations.

You may have heard the story about the difference between being involved and being committed. If not, consider a bacon-and-egg breakfast: The chicken is involved. The pig is committed. The idea of endorsement is to get everyone who may have an impact on your project committed to your plan at the beginning. All too often, project participants with a direct impact on project success, such as the customer or senior management, get the idea that they do not play a key role in creating project success and, instead, set themselves up in a role to judge, rather than create. This is a high-probability precursor to project failure. The project team must ensure that all parties who have a potential impact on project success endorse the project to the degree necessary to ensure project success. There are many ways to accomplish this. One requirement to obtaining endorsement is assuring that your team has listened to and addressed the needs of each stakeholder. Usually, you should obtain formal endorsement of both the project charter and the project work plan. In some instances, the project contract may help to fulfill one of these roles.

5.4 The Work Breakdown Structure (WBS)

The WBS provides a common framework to plan and control project work. It provides an ordered approach to summarize and drill down to more-detailed information and to provide quantitative and narrative reporting to customers and management. The WBS uses a hierarchical breakdown of project deliverables. This breakdown provides more manageable pieces of work for overall operation and control of the project.

Although the WBS is conceptually simple, I have learned that many people have great difficulty developing one. Part of the problem seems to stem from a human preference to think in terms of activities, rather than in terms of deliverables: the results of those activities, or outputs. The deliverables of a project include all of the artifacts and services that the project will produce. It is vital to provide clarity in the beginning of the project as to the scope of the facilities, equipment, and services that the project produces. (It is also important to specify what the project will not produce and what others will produce, but we will address that in Section 5.7.) The WBS is your tool to organize the entire scope of the project and to assign responsibility to produce everything.

5.4.1 TOC Approaches

TOC advocates offer two approaches to develop the project network. Most bypass the WBS. One approach, the PRT, can be similar to a WBS. The idea is to start with the end item or an intermediate objective and to ask the team involved, "What obstacles prevent us from achieving this objective?" Once you have the list of obstacles, you ask the team to state conditions that would overcome the obstacles. You then link these conditions in a logical sequence. This method ensures a coherent strategy and synchronized tactics to overcome the obstacles identified by the team. For very large projects, you could create layers of PRTs, corresponding to layers in the WBS.

Unfortunately, the simplified method for creating the PRT described by Dr. Eliyahu Goldratt in *It's Not Luck* [3] is not appropriate for generating a project WBS. The reason is that the PRT is only ensures that certain necessary conditions are met, those necessary to overcoming the obstacles that the team identifies. The method does not ensure that all deliverables sufficient to deliver the project result are included. W. Dettmer [4] describes a modified method to ensure both necessity and sufficiency.

A second TOC approach relies on backwards planning. This idea starts with the primary deliverables of the project and repeatedly asks, "What inputs are necessary to create this output?" Backwards planning follows this method until tasks are reached that do not require additional inputs.

Backwards planning has several weaknesses:

- Some people have trouble thinking in reverse order.
- Project-schedule software is built to flow down and left to right (i.e., forward planning). Backward planning makes it difficult to develop the network correctly in the software. (I like to do this using a laptop and a projector to lead the team to produce the network.)

- For larger projects, backwards planning does not provide the needed hierarchical structure to connect large networks and ensure responsibility assignment for deliverables.

Backwards planning works well for many people and does help ensure that all of the necessary tasks are included to create desired deliverables. Backwards planning can work well for small projects and is useful for developing the task detail for the work packages of larger projects. However, alone it usually does not substitute for an effective WBS, especially on larger projects.

5.4.2 The Conventional WBS

Several useful standards may aid you in developing your WBS. The PMI has a relatively new one [5], and the U.S. DOD has a very comprehensive one [6]. Harold Kerzner provides the following criteria for a WBS [7]:

> [T]he project manager must structure the work into small elements that are:
>
> - Manageable, in that specific authority and responsibility can be assigned.
> - Independent, or with minimum interfacing with and dependence on other ongoing elements.
> - Integratable so that the total package can be seen.
> - Measurable, in terms of progress.
>
> A properly prepared WBS should facilitate the following:
>
> - Ensuring better understanding of work;
> - Planning of all work;
> - Identifying end products and deliverables;
> - Defining work in successively greater detail;
> - Relating end items to objectives;
> - Assigning responsibility for all work;
> - Estimating costs and schedules;
> - Planning and allocating company resources;
> - Integrating scope, schedule, and cost;
> - Monitoring cost, schedule, and technical performance;
> - Summarizing information for management and reporting, providing traceability to lower levels of detail;
> - Controlling changes.
>
> The WBS usually has levels assigned; for example:
>
> Level 1: Total Program
> Level 2: Summary Cost Accounts
> . . .

Level n – l: Work Package
Level n: Activity

In some cases, these words have different meanings. In particular, in many cases the work package is the lowest level of work assignment, restricted to one resource provider per work package.

The WBS also has a numbering system that provides a unique number to every piece of work defined. The numbers usually follow the hierarchy of the levels, with the lowest level corresponding to a charge number for collection of cost.

Project managers use different approaches to subdivide a total project into a WBS. The most preferred is a product-oriented WBS, where each work package produces a definable, measurable output. The collection upwards then may follow functional lines, or, for major pieces of hardware (including facilities), subsystems and systems.

The most important aspect of the WBS is that it be comprehensive. Since it is the basis for all planning and cost estimating, nothing should be left out. In addition, if the project funding decision is going to be based on cost, it is imperative that the WBS not be redundant.

Many companies use templates to create the WBS for similar projects. These can be a useful resource to get started. However, templates share a major shortcoming with other checklists in that they tend to provide a degree of comfort, sometimes stifling thinking beyond the items in the checklist. The project manager has to be vigilant not to allow templates to constrain thinking and to ensure that all required work is covered in the WBS.

Sometimes clients (especially government clients) will dictate a WBS structure, usually because they need to compare projects by different contractors or according to different types of purchases. This is a legitimate client need and must be honored. The project manager still must assure that all project work is covered, that there are no redundancies, and that responsibility assignments are unique and appropriate.

5.4.3 Project Organization

Do not confuse the WBS with the project or company organization structure. Although work may align with the organization, it does not have to align. The only requirement is that at the lowest level, one individual has clear responsibility and authority for the work performed. More importantly, the WBS must define the deliverables for the project, not the functions necessary to deliver the scope.

There are many opinions about how to organize a company for project management. Since most project managers do not have the luxury of redesigning the company for their projects, I will not address the overall company organization. Project managers usually do have the flexibility and authority to design their WBSs, select their project-management teams, and assign responsibility and authority.

Following Dr. W. Edwards Deming's idea to use the overall process flow for a company as an organizing principle, an alternative I recommend is that you organize your project team around the WBS. An alternative is to make someone responsible for the critical chain and for each of the feeding buffers. Since it is unlikely that the WBS was organized this way, the project-management team may crosscut the responsibilities of the work-package managers. The project-management team has

responsibility to assure accuracy and completeness of connections between work packages and activities, the most vulnerable part of a project.

5.5 Responsibility Assignment

Responsibility assignment ensures that someone owns every element of the WBS. It used to be the fashion to create a responsibility assignment matrix. This matrix places the WBS on one side and the responsible organizational element orthogonal to it. This is a sparse matrix (i.e, only a few boxes have marks in them) if you only designate the person responsible for the specific WBS elements. For any reasonable size organization, such a matrix is too large for people to handle. This matrix is also hard to use and is difficult to keep up to date as companies change their organization.

A superior representation is the linear responsibility matrix. This matrix lists the WBS elements in the first column, the responsible person (not organizational element) in the second column, and anything else you want in the subsequent columns. This matrix is easy to develop and maintain. You can look at it on a computer screen, and you can print it on regular paper and bind it into your plans so that everyone can use it. It can also convey much more information. Table 5.1 illustrates a simple example.

You can include the WBS and responsibility assignment in most project-schedule software. Microsoft Project includes predefined columns for the WBS and a predefined text field labeled "Contact:" that you can use for the responsibility assignment. Some call the person assigned responsibility at the task level a "task manager." The task manager is responsible for delivering the required output of the task. The task manager need not be one of the resources working on the task.

5.6 Milestone Sequencing

The WBS defines the scope of the project deliverable and the key processes necessary to provide the deliverables (e.g., design), but it provides no information on the sequence of project tasks. The project plan must logically sequence all of the project tasks. For a small project (i.e., fifty tasks or fewer), you may go directly from the WBS to a task list and link the tasks using project-scheduling software. For a project with a larger number of tasks, that approach does not work. The number of task linkages rapidly becomes too large to link even a WBS-ordered task list. You need an intermediate step to facilitate generating the project-task logic.

Table 5.1 Example of a Linear Responsibility Matrix

WBS Number	*Deliverable*	*Responsible Person*	*Notes*
1	Design package	Karl Sagan	
1.1	System engineering	Karl Sagan	Lead for integration-design reviews
1.2	Hardware design	Charles Metcalf	
1.3	Software design	Simon Ligree	
2	First prototype delivered	Mary Riley	
3	System tests	John Jones	

An effective way to aid developing the logic is first to identify the major project phases in terms of key milestones. Figure 5.2 illustrates an example of the key-milestone chart structure. Each milestone must have a specific deliverable assigned to it. The milestone sequence chart does not include dates. Dates result from the integrated schedule; they are not inputs to it unless it is a project with a definitive end date, such as a proposal submission or a meeting, such as the 2004 Athens Olympics.

You may then consider what is necessary to complete each of these major milestones and build a list of supporting milestones under each of the key milestones. The resulting milestone-sequence chart, worked out jointly by all of the key project-team members, provides a basis for developing and sequencing the tasks defined in the work packages (Section 5.7). It provides many of the linkage points to tie the work packages together.

You may also use the milestone sequence as a supplemental tool for project measurement. Many organizations establish project decision "gates" as key points for project reviews, such as completion of the system engineering or of the first prototype test for development projects, or of the conceptual design for construction projects. You should expect to find these major milestones on the critical chain for the project. Management or clients often like to use milestones as indicators of project success on performance to schedule. If you put the milestones on the critical chain of the project plan, there is a very low probability that they will be completed "on time"; therefore, performance is subject to misinterpretation by people who do not yet understand the critical-chain process. In this case, I recommend adding a milestone buffer to each major measurement milestone and reporting using the end of the milestone buffer as the milestone "commitment" date. Figure 5.3 illustrates this idea. You should still control the project to the overall project buffer.

5.7 Work Packages

Work packages provide the basis for the project network, schedule, and cost estimate. They are contracts between the project manager and the work performers. They are the source documents for inputs to the integrated cost-schedule plan for

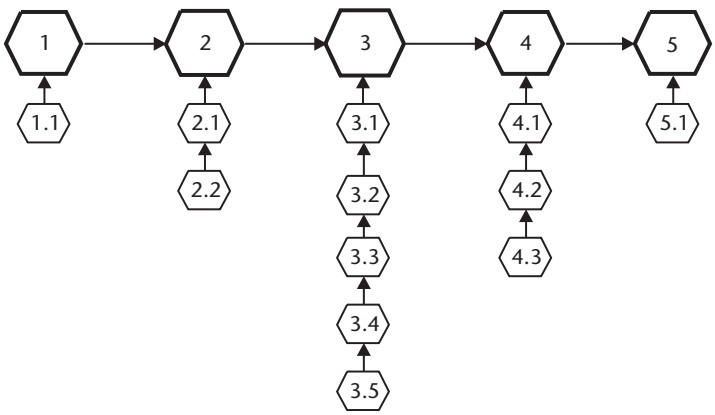


Figure 5.2 The key milestones define a backbone for the project-task sequences.

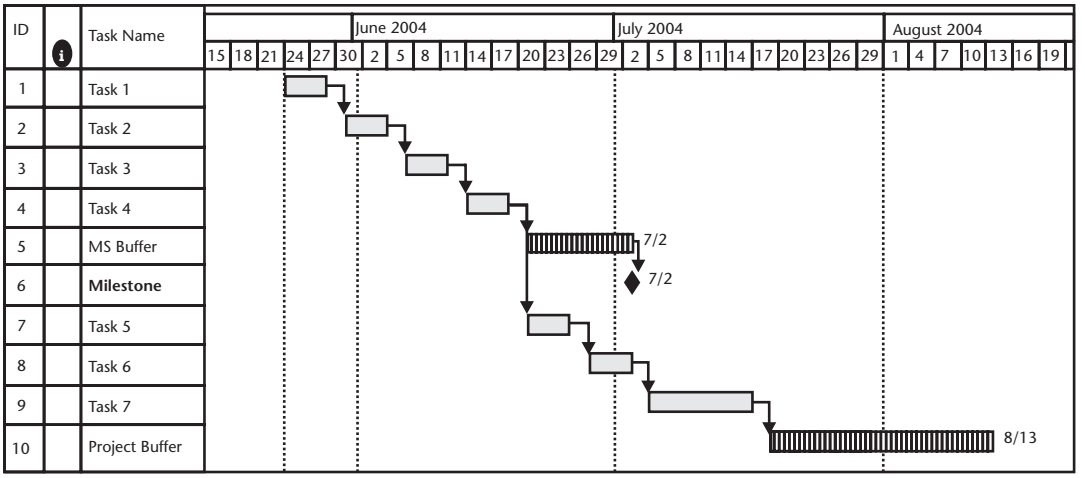


Figure 5.3 If your organization uses milestone dates to judge project progress, you must put a buffer in front of them.

the project. They contain the scope to be delivered by the work package, specifications or reference to specifications, codes and standards for the deliverables, the activity logic, activity resource estimates, and the basis for the activity resource estimates.

The design of your work-package documentation can greatly influence the ease and quality of planning the project. It is the point at which most engineers begin to whine about there being too much paper. You must design the work-package process to be simple and user friendly. Figure 5.4 illustrates the project-logic input, an essential part of the work package that, combined with the assumptions and deliverables (scope statement), delivers the information necessary for a project plan.

You must assign elements on the WBS to people to plan and manage. These individuals sometimes have a title, such as work-package manager, core-team member, or cost-account manager. They are usually technical experts in the subject matter of that portion of the WBS. They must define the detailed work scope, establish the task sequence, and estimate the task resource requirements. They are responsible for identifying the links between their work packages and others in the program. They also supply the justification for the resource estimates.

5.7.1 Assumptions

Assumptions underlie every project plan. No matter how much detail you put into the project specifications, there are always lower-level assumptions underlying that detail. There are always influences that could affect the course of your project about which you have made some (often-unstated) assumptions. Project plans should identify the key assumptions necessary to provide reasonable estimates of the project-task parameters: the resources required and task duration. For example, an assumption for a construction schedule may relate to the weather (e.g., no more than six days of outside work lost due to inclement weather). An assumption might address actions outside the direct control of the project (e.g., permit review time by regulatory agency not to exceed 30 days). You should identify these assumptions while developing the work packages.

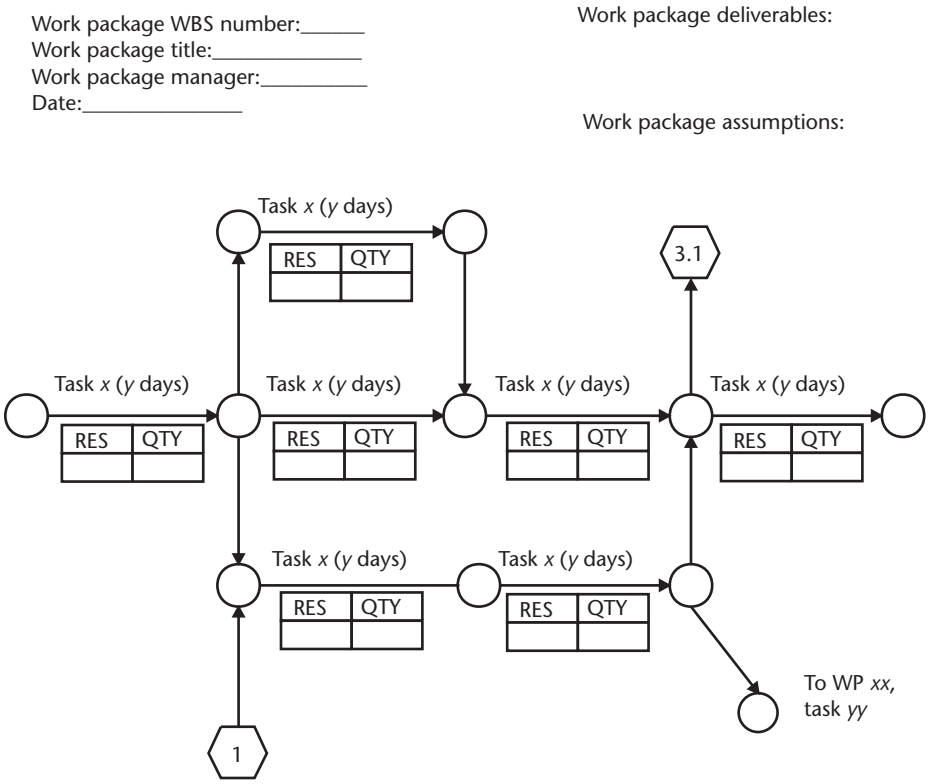


Figure 5.4 The work-package logic provides essential input to create the project plan.

You should try to counter two frequent tendencies in writing assumptions. One is to attempt to assume everything as a substitute for doing the necessary planning work. This may lead to long lists of assumptions, which you can summarize as, we're not responsible for anything. Limit your assumptions to those necessary to create an effective plan. The second frequent tendency is to write assumptions in the negative (i.e., to specify what the project will not do or what is not in the scope of the project, rather than specify specific project deliverables). Cover this instead with a positive general statement that says, "Project deliverables include only the specified items."

5.7.2 Project Network

The most important part of the work-package documentation is the network input. Note that the network input does not carry dates. It provides activity duration and logic and specifies resource requirements for the activities. One reason you should not input dates to the network is that the dates cannot be developed until the network is put together and the critical chain developed. Date constraints can prevent your schedule software from calculating the critical chain (or path), or they can distort the calculation. Also, dates assigned to individual tasks have a zero probability because all tasks are variable, and it is impossible to predict the output of one trial of a statistical variable. Your software will calculate early and late start and finish dates for each task. Then, you should completely ignore those dates. I never display the individual task-date columns.

I have seen several companies put the project-input format into the hands of budget personnel who create forms that are budget-request spreadsheets. Planners have to develop the schedule separately and figure out how to make them match. Use the work package as designed, and let the computer determine the schedule and spread the budgets for you. Task dates and budget spreadsheets are outputs from the integrated cost-schedule system, not inputs. Of course, in critical chain we are not interested in most of the task dates anyway; we are only interested in the start date of chains of tasks and the completion date of the project.

5.7.2.1 Project Logic

The project logic defines the necessary sequence of tasks to achieve the project result. Work packages are simply small projects: each requires its own logic. You must link tasks so that the output of one task comprises the input of the next task. You can think of each project task as a little work process, with inputs and outputs. Project tasks conform to the TQM supplier→input→process→output→customer (SIPOC) model. The supplier is the task manager of the predecessor task. The customer is the task manager of the successor tasks. You then link work packages using milestones and other task links.

The most common task logic links the finish of one task to the start of the next (output to input). The most common relationships, or links, available in most schedule software are

- Finish-to-start: Often called the predecessor-successor relationship, this clearly illustrates how the output of one task is required as the input to start the next task in the sequence. Most of your network tasks should use finish-to-start relationships.
- Start-to-start (with a lag): Use this relationship when two tasks can be carried out simultaneously, once the first task has created some amount of output for the second task to work on. For example, you may have to create many copies of something that requires three steps. Rather than schedule all three steps for every copy, you put in one task for each of the steps, titled something like "Step 1 for 100 copies of *x*," "Step 2 for 100 copies…," and so forth, with each step lagging by the amount of time necessary to complete the first item.
- Finish-to-finish: Use this relationship when things must finish at the same time but may have different start times. For example, you need a number of chapters for a book to have all of their cross-references match.

Computer schedule software frequently offers a host of other possible constraints, including fixed-date constraints. The software also allows you to specify leads or lags on the relationships connecting tasks. You should apply these other constraints as little as possible, as people often lose track of them, and they can cause unexpected results when calculating a project network.

Some critical-chain software only allows finish-to-start relationships and/or one or two other types (e.g., start no earlier than). They may not allow leads and lags. If special situations demand it, you usually can model a situation that meets your needs with only regular tasks and finish-to-start relationships, although it may require some dummy tasks to do so. Be sure to understand the capabilities of your software.

Except for fixed-date requirements (e.g., a project with a specific date, such as an Olympic stadium) and nonnegotiable input dates (e.g., available funding), dates should be an output of your network calculation, not an input to it. You should not date-constrain the individual tasks in your network. You should link them and let the software do the calculation. *Remember: the dates assigned to individual project tasks by the software all have a probability of zero*. Individual task dates are meaningless.

You should check your project logic, considering the following points:

- Does each task have a clearly defined output?
- Are the predecessors to each task necessary to start the task?
- Are the predecessors to each task sufficient to perform the task?
- Do the tasks (collectively) provide for all of the project deliverables as outputs (compared to the WBS)?
- Do the tasks specify the necessary resources?
- Do the tasks have unnecessary date constraints?
- Are all of the milestones on the milestone-sequence chart included?
- Are the resources that determine task duration working at 100% utilization?
- Do all of the project-network paths tie in to the end of the project? (If not, tie them in at least to a milestone for "Project Complete.")

Do not link summary tasks in a project plan. Link the subtasks underneath the summary tasks. The next chapter describes how to create the critical-chain plan from this resource-loaded project network.

5.7.2.2 Resource-Loading Your Network

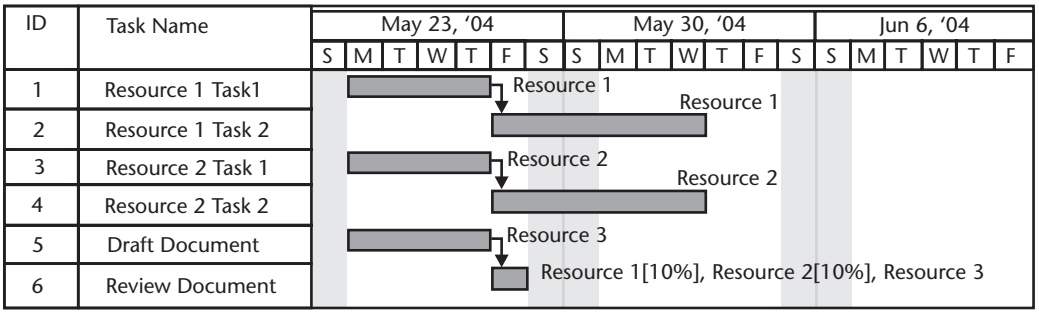
Once you have determined the tasks required to produce your project deliverables (the network), you must specify the resources required to make each task output. Although conventional project management has always allowed for, and in many cases encouraged, integrating the resource requirements with the schedule network, critical chain absolutely requires it. Critical chain is going to adjust the duration of the project-task network to match the demanded resources to those you tell it are available.

You have to make several decisions when assigning resources within your schedule software. First, you have to decide what resources you are going to identify. You usually have to identify the human resources needed for the tasks, but most of the scheduling software does not require that a resource be a person: it can be a machine (e.g., a crane), or it can be any other constraint (e.g., the hatch on a submarine or a space limitations on how many people can work in a certain area). You should only include those that are likely to be limiting to your schedule.

For human resources, you have to decide if you are going to identify individuals by name or by skill type. If you have a small group working on your project, you can identify the specific individuals that will do the work. Larger organizations and projects may have many individuals who can do certain kinds of work. In those cases, you increase flexibility (and may accelerate the project) by assigning resources

by skill type and specifying how many resources of that skill the task can reasonably use. For reasons that will be clarified later, you should generally put down the maximum number of people who can efficiently work on a task and reduce the estimated task duration accordingly.

The critical-chain relay-racer task approach encourages you to fully dedicate the resource(s) that determine the overall duration of a task to that task. You may assign other resources at a fractional level, but be aware that this may cause unexpected results when you resource-level your network. Your digital computer will resource-level to an algorithm with specific rules. Some of these rules may be user adjustable, but once set, the software applies them ruthlessly. Thus, if you load a secondary resource at 49% to two tasks and have 100% of the resource available, the scheduling software may schedule them in parallel. If you assign the resource at 51%, it may delay one of the tasks completely until the other is done. (Some of the scheduling-software algorithms enable leveling over user-selectable "buckets" of time, for instance, days, weeks, or months. Thus, whether the surface moves a task to it level resources can depend on the task length and location in the schedule, as well as the resource loading.) Figure 5.5 illustrates a potential unintended consequence of resource leveling. Something like this actually happened on a critical-chain project, and the technical people claimed, "The logic is wrong!" The logic is correct, and the computer did exactly as instructed. In this case, they probably should not have resource-loaded the review task unless it was a full-time task assignment for the resources. In most software, you can include task notes to specify who should review a deliverable. Usually it takes some experimentation and experience to determine how best to model resources in your particular project plans.



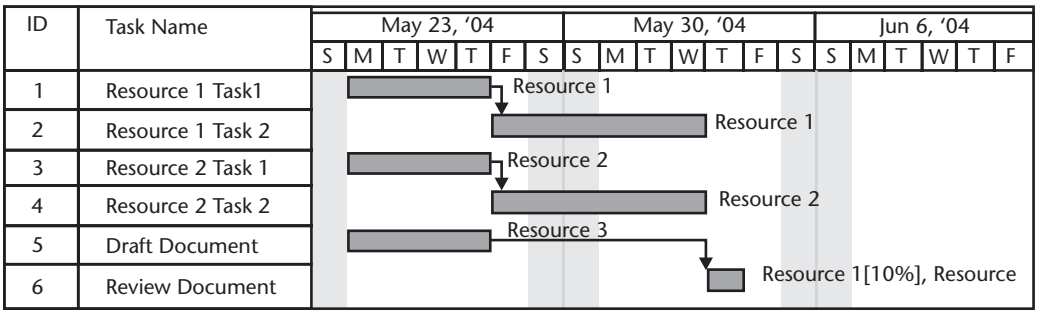


Figure 5.5 Example of resource leveling causing an unexpected result: (a) before leveling; (b) after leveling: the document review gets displaced to the right due to 10% loading of resource.

5.7.2.3 How Many Tasks (a.k.a. Granularity)?

Many project plans contain thousands of individual tasks. Some contain tens of thousands. Guidance on how many tasks to include in a project network usually suggests breaking down tasks into minor durations to facilitate project reporting by task completion, rather than using estimates of percentage complete or time remaining. Consideration of the purpose of the project plan and the TOC leads to the recommendation that project plans be only as detailed as you need to run your project successfully and no more. Project tasks generally should represent a hand off of a specific artifact from one resource (or group of resources) to another.

Project plans are not a substitute for detailed design information. You can link such information to your schedule by a number of means, including using the WBS as a file structure for such information, task notes, and hyperlinks in your schedule software (depending on your software). Avoid the temptation to use your schedule software for other than schedule management. Your highest-level project plans should only rarely exceed a few hundred activities. Larger projects require a hierarchy of plans, where no individual plan contains more than a few hundred activities. In these cases, you should link from lower-level plans to the higher-level plan at the start of the lower-level-plan project buffer.

The primary reason to limit the number of tasks in a project plan is that overall uncertainty does not justify too much detail. Too much detail increases the work to create and maintain the plan, as well as the probability of errors in the plan. Very few people can understand a plan with more than a hundred activities, even with study. Since the number of potential links in a plan increase exponentially with the number of tasks, it is highly unlikely that a plan with even a hundred activities will be error free.

Consider the fact that the average size of a task (in terms of dollars, total path time, or total resource time) is the inverse of the number of tasks in the plan. Therefore, the average size of a task in a plan with one hundred activities is 1% of the total project. Since most of the tasks can be estimated with accuracy no better than ±50 to serveral hundred percent, it makes little sense to divide the project into smaller and smaller pieces.

People often suggest that an insufficiently detailed plan is a cause of project failure. When projects "fail," they usually do so in the range of several hundreds of percent, (i.e, cost two or three times the initial estimate) not fractions of one percent. It is not logical to conclude that plans with one hundred or more activities are not detailed enough to prevent project failure. You are not likely to find missing pieces that add up to hundreds of percentage points by subdividing chunks that average only 1% of the total. The problem is elsewhere, and so is the solution.

More tasks increases the amount of effort required to develop the project plan. A given planning effort spread over more tasks means less effort per task. This may lead to a less accurate plan, not a more accurate plan. If you have the ability to put in more planning effort, you should apply it to looking at the spaces between the tasks, ensuring that all of the inputs and outputs are correct and considering the resources and processes within the tasks, rather than adding tasks to the project plan.

For statistical reasons, there is value in ensuring that the critical chain of your plan contains at least ten activities. This increases the chances that statistical fluctuations will tend to offset each other. Also, no single activity duration should

exceed 20% of your critical chain. If one task dominates the critical chain, you are more subject to variation in that individual task, and you are more vulnerable to inaccurate estimates of the time to complete. Consider defining intermediate deliverables to divide a dominant task.

On the other hand, if you have many tasks on a path, and several tasks in sequence use the same resource, consider combining those tasks and defining the final deliverable as the task output.

The above considerations (number and relative size of activities) apply to both feeding chains and the critical chain, but they are a little less important on feeding chains because the schedule protects feeding chains with both the feeding buffer and the project buffer.

5.7.3 Activity Duration Estimate

Chapter 3 demonstrated the importance of the activity-duration estimate. When starting with critical chain, start where you are: *solicit task duration as you have always done*. Do not ask for the average duration for each activity. The reason is that people do not have an intuitive sense of *average* and will tend to give you an estimate they are comfortable with, no matter what you choose to call it. If you ask for the *average*, you will then have trouble getting a shorter estimate to represent the average.

You should ensure that all work estimates include 100% effort on the task. If they do not, reduce the task duration, keeping the work (person-hours or person-days) the same. In other words, if the task has an engineer at half time for 10 days, reduce the duration to 5 days with the engineer working full-time.

Next, you need to allocate some of the allocated task durations to the tasks in the plan, and some to the buffer. People have various ways of doing this. The simplest one, initially advocated by Goldratt and still very powerful and useful in many environments, is to estimate the mean task duration (used for the task duration in the network) as one half of the usual estimate. An exception is made for tasks that have a finite cook time that cannot be reduced (e.g., the gestation period of mice). The other half of the duration will go into the buffer, through one of two methods. The first method, again suggested by Goldratt, is to size the total schedule buffer as one half of the task time of the chain it is protecting. The other approach uses statistical theory and will be covered in Section 6.4.

Another approach is to solicit mean estimates from your estimators. You should only attempt this approach *after* you have obtained the initial "normal" estimates. You should go back and request "average" estimates, using a question like, how quickly could you perform this task if you had all of the inputs you needed at the start and if everything went right? If you do not get a significantly reduced estimate, you must work with the estimators to understand their reasons. You need a substantial difference between average and low-risk estimates (e.g. 1/2) to generate the project buffer. Adding a project buffer to low-risk estimates needlessly extends the project schedule, and will cause performance to be less than achievable.

5.7.4 Uncertainty Revisited

Project managers face a conflict over the uncertainty of estimates. Pressure comes from management or clients, who will say, "If your uncertainty is over x percent,

you must not have done a good job estimating." Human beings are by nature inclined to overconfidence in their predictions.

The following discussion treats uncertainty in cost and task duration as if they are interchangeable. This discussion considers work performed by people or resources (e.g., rented machines) charged on a unit or time basis; that is, cost equals the rate times the amount of use. Therefore, uncertainty in the cost percentage is the same as the uncertainty in the use percentage. Likewise, because the duration is the use rate times the time, the uncertainty in the duration percentage is directly proportional to the uncertainty in the cost percentage.

Since projects are by definition one-of-a-kind and first-of-a-kind, we often lack statistical information to quantify uncertainty in estimates or task performance. Consider what we do know about uncertainty. Ask someone to give you an estimate on a new house. He or she might start by saying something like, "What are your specifications?" Permanent houses in the United States today can range in price from $80,000 to millions of dollars. The most important question on a house is, where is it? Second most important is, how big is it? Even with those specifications fixed, prices per square foot can range over a factor of two, depending on the type of construction, interior finishing, and so on. Then, of course, the price can vary by at least 10% for houses with identical specifications, location, and condition, depending on how much the seller has invested, how good a negotiator the buyer is, the general market in the area, the seller's motivation, and other factors.

Therefore, we cannot get too close in our estimate on a house. How much does a car cost? Same routine. Even an identical new car can vary by at least ±10% for two purchasers in the same town.

One of the best-selling project-management books (which I choose not to reference out of charity) says, "The first type of estimate is an order-of-magnitude analysis, which is made without any detailed engineering data. The overall analysis may have an accuracy of ±35% within the scope of the project." (And I thought *order-of-magnitude* meant a factor of ten. Alas.) After a few intermediate steps, the text states, "The definitive estimate, also referred to as detailed estimate, has an accuracy of ±5%." Wait a minute. We just agreed above that the actual cost of a very well-known, existing automobile (on the lot, so to speak), i.e identical items, at the same time and place, can vary by twice that amount. How could we possibly expect an estimate of a less-known entity to have twice the accuracy?

One source claims in another table that for low-risk projects, work-package estimates have overall uncertainty of 2%, the subtasks of 5%, the task of 10%, the project of 20%, and the program of 35%; that is, this source claims that as we combine individual estimates of lower uncertainty, we get a higher overall uncertainty. This source has repealed the laws of statistics. (Perhaps this table was printed upside down.)

It is interesting to find through review of many project-management books that the same cost-estimating-accuracy estimates keep appearing, but they are never referenced to source material. The only source material referenced relates to construction-cost-estimating guides, which provide some accuracy estimates quite inconsistent with the project- and program-cost estimates stated. For example, the 1997 *National Construction Estimator* [8] states,

> Estimating is an art, not a science. On many jobs, the range between high and low bid will be 20% or more. There's room for legitimate disagreement on what the correct costs are, even when complete plans and specifications are available, the date and site are established, and labor and material costs are identical for all bidders.

Obviously, other projects, such as R&D or information-technology projects, can have much higher uncertainty than construction projects with detailed specifications.

Finally, I could find no books providing an operational definition of the meaning of the term *accuracy* as ±35%. I may think this is the standard deviation; you may think that this is the "extreme value," or 99% probability number, assuming a normal distribution, which is probably incorrect for cost estimates. (If my understanding is correct, it means that according to your definition, the accuracy is ±115%. Well, perhaps not minus that amount!) Can we be sure we understand the word the same way?

We all know of several large projects that have overrun their initial schedule and cost estimates by two to three times (see Chapter 1 if you need some reminders), and perhaps even of some that spent all of the project money and then were canceled, with nothing to show for it. While multibillion-dollar government projects inevitably come to mind first, plenty of large commercial projects have the same performance history. Does this mean that there is a systematic bias to underestimate?

Research by major construction firms and experience with critical-chain projects demonstrate that projects that complete on schedule usually complete within or near the original budget. This may not be true if the schedule was maintained through extensive overtime, although CCPM projects continue to show reductions in overall overtime while accelerating schedules through focusing overtime only on the tasks that matter to the schedule. Projects that overrun do not begin to see that they are in trouble until a significant portion of the original plan is expended (in money or time), usually one half to two thirds of the original estimate. This is the phase of a project in which the expenditure rate is at a maximum. Extending the project duration at this maximum rate creates a disproportionately large impact on project cost. Thus, you should not directly relate these reported cost overruns to schedule runs.

If the project behavior is as we hypothesized in Chapter 2, that is no credit for positive variances; it only takes one late task on the critical path to make a project late. In addition, since all of the money is spent on the tasks in keeping with the "use it or lose it" philosophy, the schedule extension should lead to a cost overrun. Based on the above, it may lead to a very large overrun. These large cost overruns are not, therefore, part of the new critical-chain paradigm since the critical-chain method explicitly removes the sources of the overruns.

You should, therefore, demand significant differences between the "low-risk" estimate to perform a single task and the "average" estimate for that task: a factor of two or more. People who suggest that these differ by only a few percent likely do not understand the reality of variation. Explore the basis for that estimate. You should find that the average task-duration estimates are approximately one half to one third of the low-risk estimates for the individual task duration.

5.8 Need for Cost Buffer

If cost is important in your world, your project work plan should include a cost estimate, including a cost buffer. You should size the cost buffer for the project considering the project risks and accuracy of the estimates. If you are using project software for your cost estimate, keep in mind that you have estimated each task at its 50–50 probability. You should expect to use a significant part of your feeding and project buffers. You have to include an estimate for this use. This is your "cost buffer."

You are better off using a single, aggregated project cost buffer for the same reasons you are better off using a single, aggregated project time buffer. You get the statistical advantages of independent estimates and the psychological advantage of not having it associated with specific tasks.

I describe in detail in "Schedule and Cost Buffer Sizing: How to Account for the Bias between Project Performance and Your Model" [9] the two kinds of uncertainty you are accounting for:

1. Bias, which could sum for all of the activities subject to the bias;
2. Statistical fluctuations, which will sum as the sum of the squares for all of the independent cost elements.

Bias includes the fact that the variations in cost estimates tend to be skewed distributions; that is, most work-package managers have been trained to spend all of the money in their work package, or they will get less next time. Your estimate of how successful you are at changing this paradigm should influence your estimate of the cost buffer.

M. R. Vigder and A. W. Kark [10] performed a recent study on software projects and note,

> A number of the projects we investigated were large-scale systems, involving more than four years duration and more than 100 person-years of effort. Without exception, the costs of all of these projects were seriously underestimated.

They note the following as some of the reasons:

- In large-scale systems, complexity does not increase linearly with lines of code, but rather exponentially.
- The larger a system and the further into the future its delivery, the more difficult it is to correctly and completely specify all the requirements. These cost overruns seem to be amplified for large systems.
- The longer the duration between initial requirements and delivery, the more likely there are to be changes in the requirements. This can occur due to changing user expectations, changes to the environment in which the system is to be installed, or new personnel with different views on what the requirements should be involved.
- Long project duration means that technology advances may outstrip the initial requirements.

Most of these factors do not seem to be confined to software projects. You should consider them when estimating the buffer size to cover bias in the estimates. Section 6.5 describes cost buffer sizing.

5.9 Basis for Cost Estimates

The cost-estimate basis is part of the work-package documentation. It is an extremely important element of planning for cost-plus contracts and for many government contracts. It is also the subject of most difficulty for many engineers. They have no trouble estimating resource needs; they just cannot seem to tell you how they came up with that particular number, other than providing a meaningless and insufficient phrase, such as "engineering judgment" or "past experience." You might start by asking what assumptions they used to come up with the estimate.

Professional estimators have no trouble coming up with the estimate basis. You usually get it without asking for it. They will refer to guides, previous experience (specific), and quotes from vendors, or otherwise substantiate the numbers used. It is usually quite simple to provide the basis for any kind of hardware (e.g., 500 feet of 4 inch pipe 1 ~ $1.85/ft [ref. Joe's plumbing telephone quote to Jim A. on 3/15/96]). There are books of cost-estimating factors for routine construction, software coding, and other specific types of skills. The point is to define the estimate basis well enough so that later on, if changes are proposed, you can clearly define what was in the initial scope and what was not.

5.10 The Project Work Plan

The project work plan, sometimes called the project plan, project execution plan, or project-management plan, puts the elements developed above into a form accessible to all of the project participants. It is the key to communication within the project. The PMBOK™ Guide [1] notes that the project plan is used to

- Guide project execution;
- Document project-planning assumptions;
- Document the planning decisions regarding alternatives chosen;
- Facilitate communication among stakeholders;
- Define key management reviews as to content, extent, and timing;
- Provide a baseline for progress measurement and project control.

The project plan may include, or refer to, a number of other elements for larger projects, including

- Topical plans, such as quality, safety, procurement, staffing, environmental, or systems engineering;
- Project communication guidelines and project reporting, including document distribution and approvals;

- Work procedures;
- Specifications and standards;
- Change-control procedures.

For larger projects subject to changes, you must place the key elements of the project work plan into a system that ensures people only work to the latest approved version of the plan. Many companies accomplish this today by creating the project plan as an intranet page or using a file-sharing approach.

5.11 Change Management

Change management ensures that only changes approved by the project manager are implemented on the project. The most important function of change control is to ensure that everyone working on the project is working to the same plan, including the same scope of work and detailed project requirements. Other functions of change control include

- Ensuring that people only work on approved changes;
- Assessing the impact of changes on cost or schedule before deciding to implement them;
- Billing the customer for customer-directed changes;
- Providing a record of changes;
- Providing traceability to the original project baseline.

For larger projects, change control may be part of your project-quality system. For smaller projects, it may be a memo from the project manager approving a change and identifying the latest version of the specifications and project plan.

5.12 Project Closure

Project plans often neglect closure of the project. Project closure includes dealing with the entire project's administrative, facility, and personnel issues as the project is finally completed. It usually involves final billing, disposition of project records, and closing the project office. For organizations that perform multiple projects, it should also include a "lessons learned" assessment to improve processes on future projects.

5.13 Summary

This chapter has provided the process and tools necessary to create an effective project work plan. The key elements I have identified necessary for all projects are:

- The project charter is a necessary precursor to a successful project plan that effectively meets all project stakeholder requirements.

- The WBS logically defines the general project work scope and provides the framework for responsibility assignment.
- The stakeholder-endorsed project work plan defines the scope, schedule, responsibilities, and budget for the project.
- Project networks should be as simple as possible to perform the project.
- The project plan requires a correct, resource-loaded logic network to develop the schedule.
- Dates are outputs from the logic network, not inputs.
- If cost is important to your projects, you should include a cost buffer in the cost estimate.
- You should initially request task-duration estimates as you have in the past, then apply one of several methods to allocate the overall duration to the task and the buffer.
- Most projects require a change-control process.
- All project plans should consider project closure as part of the plan.

You should adjust the degree of detail you put into the project plan and degree of formality you put into the project documentation to match stakeholder needs. In general, larger, longer, and government projects require more detail and more formality. Less experienced teams may also require more documentation and training.

References

[1] PMI, *A Guide to the Project Management Body of Knowledge, 2000 Edition*. Newtown Square, PA: PMI, 2000.

[2] CH2MHILL, *Project Delivery System: A System and Process for Benchmark Performance*, Denver, CO: CH2MHILL, 1996.

[3] Goldratt, Eliyahu M., *It's Not Luck*, Great Barrington, MA: North River Press, 1994.

[4] Dettmer, H. William, *Goldratt's Theory of Constraints*, Milwaukee, WI: ASQC, 1997.

[5] PMI. *Practice Standard for Work Breakdown Structures*. Newton Square, PA: PMI, 2001.

[6] U.S. Department of Defense, *DoD Handbook—Work Breakdown Structures*. MIL-HDBK-881, 1998, available at http://dcarc.pae.osd.mil/881handbook/milhdbk 881_cover_chap1.pdf (accessed May 24, 2004).

[7] Kerzner, Harold, *Project Management, A Systems Approach to Planning, Scheduling, and Controlling*, 4th ed., New York: Van Norstrand, 1992.

[8] Kiley, Martin D. and Marques Allyn, *1997 National Construction Estimator,* Carlsbad, CA, Craftsman Book Company, 1997.

[9] Leach, L. "Schedule and Cost Buffer Sizing: How to Account for the Bias between Project Performance and Your Model," *Project Management Journal*, Vol. 34, No. 2, June 2003.

[10] Vigder, M. R., and A. W. Kark, "Software Cost Estimation and Control," NRC-CNRC (National Research Council Canada), NRC No. 37166, February 1994.

CHAPTER 6

Developing the (Single-Project) Critical-Chain Plan

This chapter first presents the overall process to create the single-project critical-chain plan and then takes you through examples and exercises to practice the ideas. You should understand the ideas before you begin to use computer schedule aids to develop your critical-chain project plans.

Keep in mind that your schedule is just a model of reality. As with all models, it is not reality. It just needs to be good enough to help you execute the project in a way that produces the project result as soon as possible.

6.1 Process

The basic steps of the process to create a single critical-chain project schedule follow. I have written this procedure assuming that you are not using a critical-chain computer-scheduling tool. Project managers have used it to successfully plan and run single CCPM projects of up to $5 million. Larger projects or projects in a multi-project environment require critical-chain-capable scheduling software.

Working some networks by hand helps you to understand the problem that the computer is solving. This will aid greatly when you are diagnosing unexpected results.

1. Identify the critical chain.
   - 1.1. Lay out the late-finish network of tasks. The tasks must identify the mean time-duration estimate (50–50 time) and primary resource requirements. (For tasks with multiple resources, identify the primary resource you believe will be a constraint. If there are several constraint resources, break the task up for each primary resource.)
   - 1.2. If you do not have resource contention in your project, go to step 1.6.
   - 1.3. Identify the contention you will resolve first. This should be the contention nearest to project completion or the one that shows the most conflict. If several show about the same amount of potential conflict, choose the first one you come to working backwards from the end of the schedule.
   - 1.4. Remove resource contention by resequencing tasks earlier in time. (Do not worry about creating new conflicts with this step; you will resolve those in sequence.)

1.5. Return to the end of the schedule and follow step 1.4 for the next resource. As you resolve conflicts for the next resource, you must maintain the lack of the conflict for the resources you resolved earlier. Repeat until all identified resource types are resolved.
   1.6. Identify the critical chain as the longest chain of dependent events.
2. Exploit the critical chain.
   2.1. Review your plan to determine if resequencing can shorten the overall project duration. If so, do it. Do not trial-and-error various solutions.
   2.2. Add the project buffer to the end of the critical chain.
3. Subordinate the other tasks, paths, and resources to the critical chain.
   3.1. Protect the critical chain by adding feeding buffers to all chains that feed the critical chain. Size these buffers using the longest preceding path. (Note: All noncritical chains feed the critical chain to complete the project. If chains go directly to the project buffer, they also need feeding buffers.)
   3.2. Resolve any resource contentions created by adding feeding buffers through resequencing tasks earlier in time.
   3.3. Move to an earlier time any dependent tasks preceding those moved.
4. Elevate (shorten) the lead-time of the project by using added resources for certain windows of time to break contention.
5. Go back to step one. Do not allow inertia to become the constraint.

A critical-chain plan only schedules (i.e., assigns dates) to the start of the chains and the completion of the project. You should avoid publishing and discussing individual task-start and -complete dates. They are meaningless. For these reasons, you may consider talking about the critical-chain plan, rather than the critical-chain schedule.

6.2 Good Enough

"Good enough" is an important idea in developing critical-chain project plans. For mathematical reasons, it is impossible to build a precise optimizing algorithm for resource leveling. The procedure to develop the critical-chain plan ensures that the plan you build will be "good enough." This means that the overall length of the schedule will be, within a small part of the length of the project buffer, nearly the shortest or optimum schedule path. Since reality will change many assumptions, and we cannot explicitly predict the results of statistical fluctuations, this is good enough.

6.3 Examples and Practice

6.3.1 Small Example

Figure 6.1 illustrates a small example to work into a critical chain. The figure illustrates the plan in a conventional critical-path display with the early-start schedule. The first number on each bar is the WBS task identification. The second number in

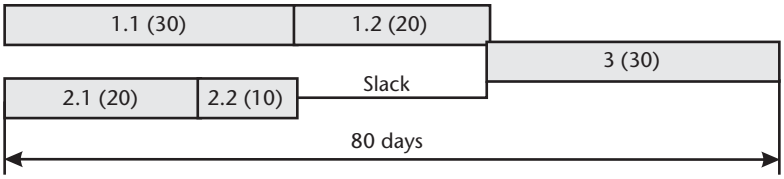


Figure 6.1 A simple project illustrates a normal early-start schedule.

parenthesis is the task duration in days. Note that task 3 depends on completing tasks 1.2 and 2.2.

Following the procedure, we first cut the task times to the 50–50 estimate and push all of the tasks to the latest time possible, considering the network dependency (Figure 6.2).

Next, we add the project and feeding buffers. Since all tasks use the same resource, we do not need to add resource buffers to this project (Figure 6.3).

Now consider the same little project with resource contention. Figure 6.4 shows the unscaled network of tasks with a PERT chart representation of your project. The network shows the different resources as colors. You can think of the colors as different skills (e.g., engineers, musicians, or equipment operators).

We lay the network out, with all of the tasks pushed as late as possible. In this case, all we have to do is add a "start as late as possible constraint" to task 2.1.

Next, remove the resource conflict, working backwards from the end of the project (Figure 6.5).

Figure 6.6 illustrates the two ways we could resolve the green resource contention. Note that each schedule shows a new dependency for the resource constraint.

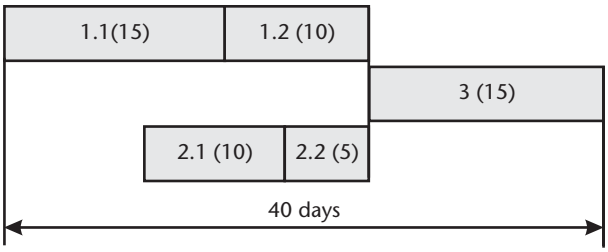


Figure 6.2 The first step to create the critical chain reduces the task times and organizes tasks to a late-finish schedule.

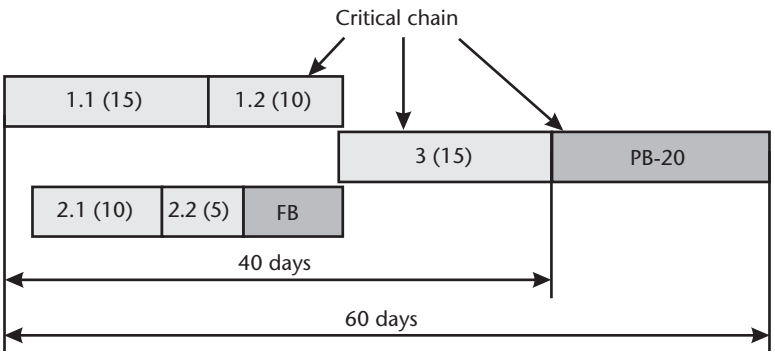


Figure 6.3 With no resource contention, just add buffers!

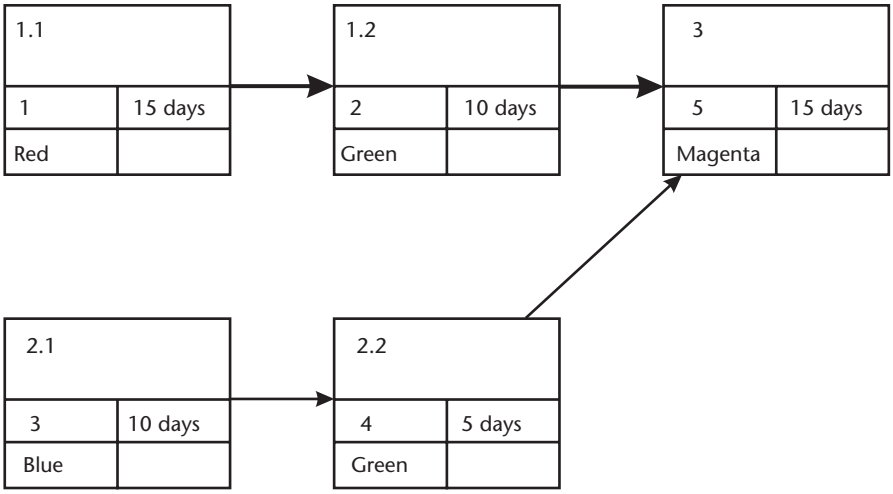


Figure 6.4 The same projects with specific resource assignments.

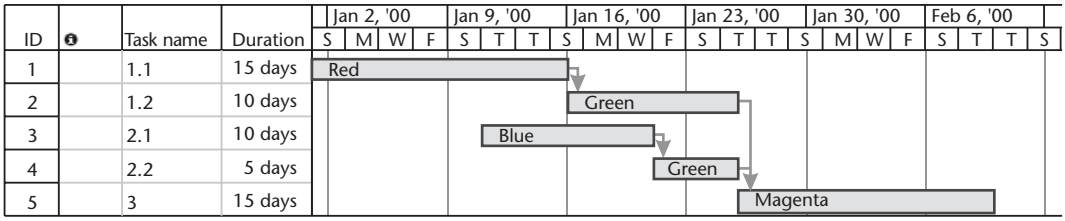


Figure 6.5 The first step pushes tasks to the late finish.

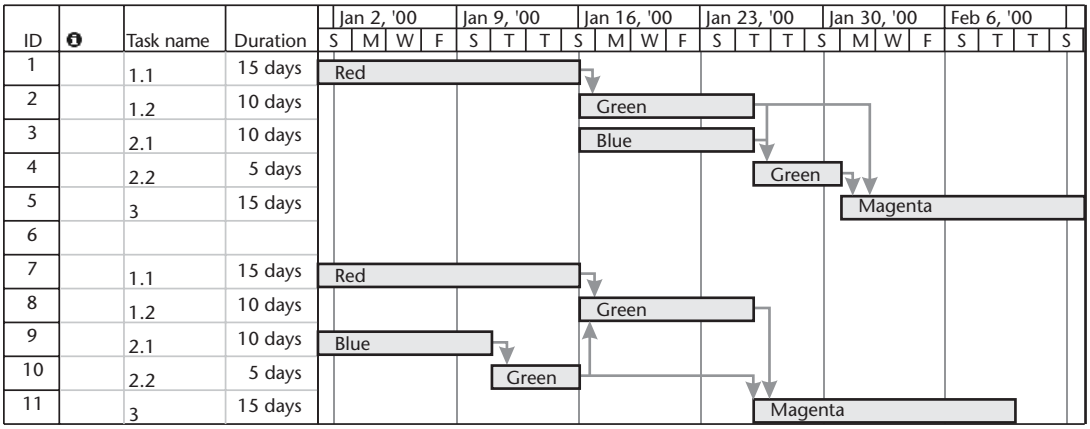


Figure 6.6 Alternative ways to resolve resource conflict.

Which resolution choice is better? You may initially think that the lower choice is better because it is a shorter schedule. With the lower choice, the two chains are exactly the same length, so we can choose either one as the critical chain. Add the project buffer and critical-chain feeding buffer to each option, and see what happens.

Figure 6.7 illustrates the critical-chain plan for the first choice of removing conflict for the green resource. The critical chain comprises all of the project tasks

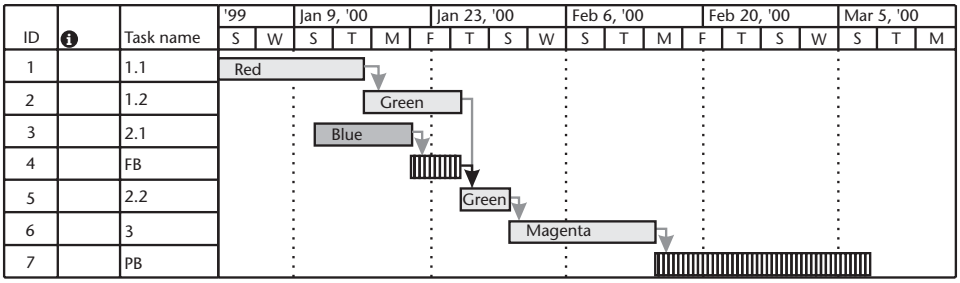


Figure 6.7 The first choice results in a planned completion date of March 7, 2000.

except task 2.1. Size feeding buffers as 50% of the total duration of the tasks in the feeding chain.

The two choices of critical chain from the first resource-resolution option both lead to the same overall length of schedule. Both also create the situation where the noncritical chain is longer than the critical chain after adding the feeding buffer. This is okay; we just have the extra lead-time for the non-critical-chain paths. Two or more nearly equal length paths may cause this, but you usually will not notice it on larger projects. Continue on to remove the next contention (Figure 6.8).

Any of the above paths are suitable for the plan because of the small differences compared to the project buffer. It often works out a little better to resolve contention by moving the longer of the two or more tasks backwards in time. This tends to keep the critical chain as the longest chain, therefore increasing the project buffer, helping to protect your project from common-cause variation.

Lay out the Figure 6.9 exercise as a critical-chain plan, with all of the appropriate buffers. The first line in each box represents the task number. The second line represents the resource, by color. The third line represents the (already-reduced) task time, in days. (See the last question in Section 6.10 for the approximate length of the schedule you should have obtained. Note that a good-enough schedule is within a small part of the project buffer.

6.3.2 Large Example

Figure 6.10 presents the task network for the large example. The top line of each box is an identifier for the task. The color relates to a specific resource. The numbers at the bottom of each box represent the task duration in days. The task durations have already been cut by 50%.

Lay out the critical-chain plan for this project.

The first step is to lay out the network. Figure 6.11 illustrates the project entered in the MS Project Gantt Chart view. Looking from left to right, note how the magenta (Mag) tasks A-1 and B-2 overlap. Then the blue resource tasks B-3, C-3, and D-3 overlap. It is easy to see that the critical path through this network is the top logic path, as it shows no gaps.

MS Project enables you to remove these contentions by resource leveling, as illustrated by Figure 6.12. Note that most chains now have gaps in them, making it impossible to define a critical path.

Next, you must identify the critical chain. Figure 6.13 illustrates the critical chain identified by the CCPM+ software. Observe how the chain jumps the logic

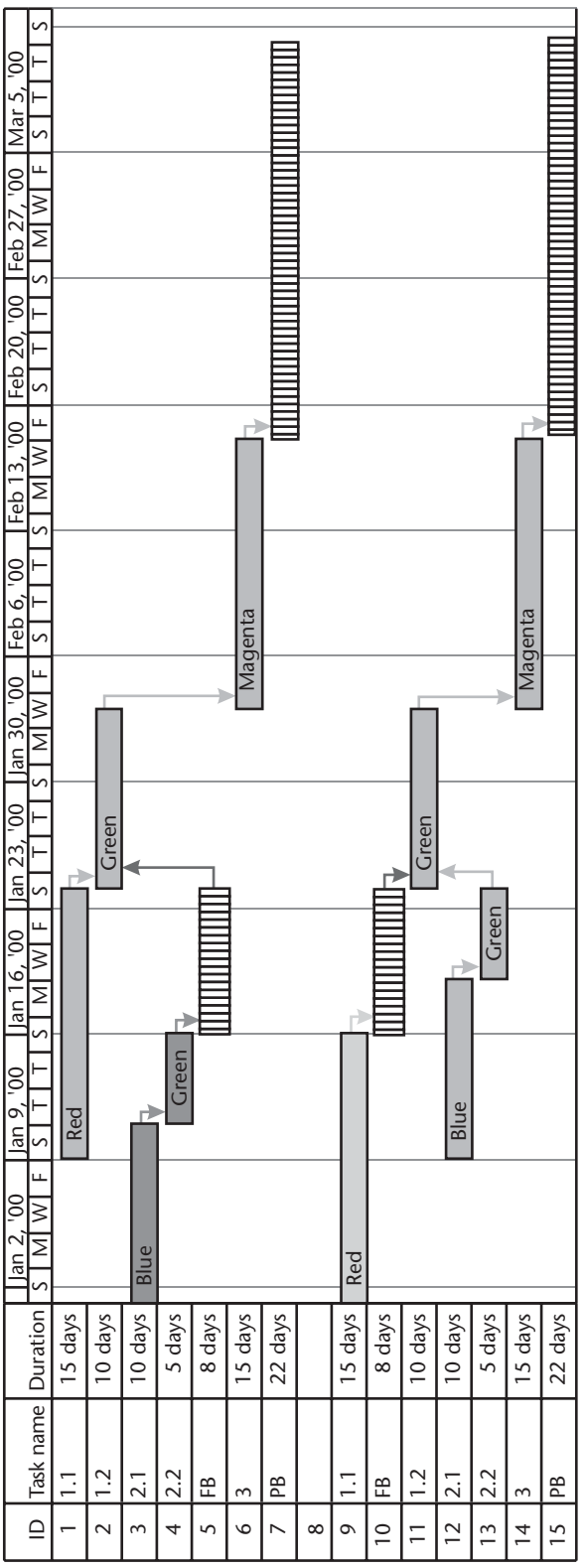


Figure 6.8 The second resource resolution choice leads to the same lead time for the two choices of critical chain and completion on March 10, 2000.

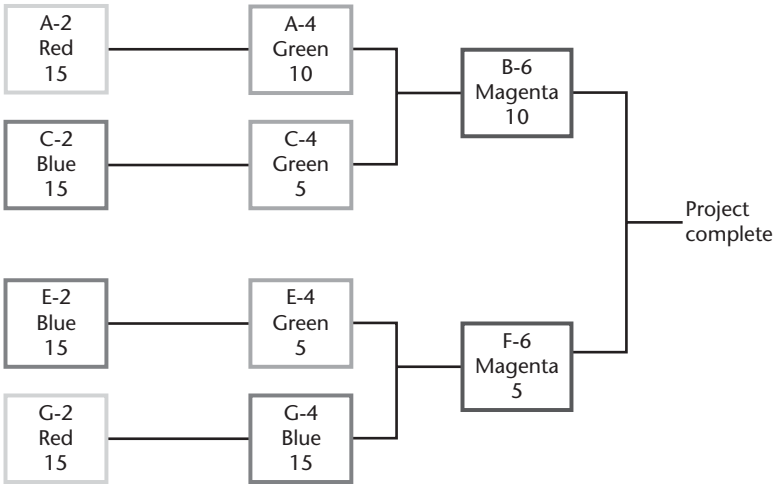


Figure 6.9 Small exercise.

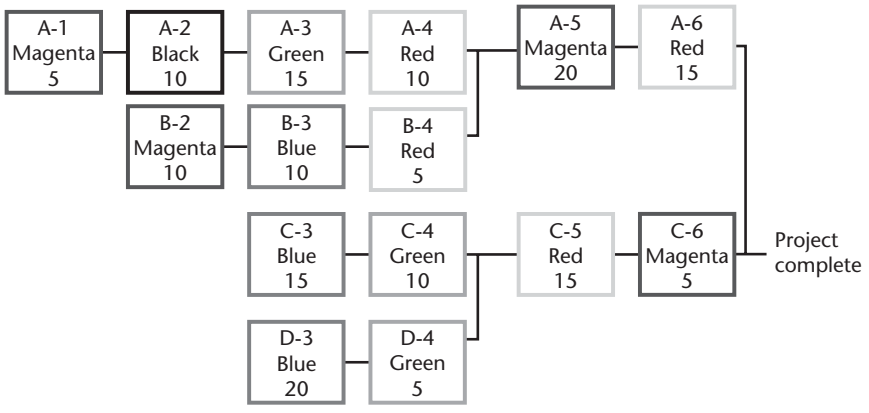


Figure 6.10 Large example. (Task times already reduced.)

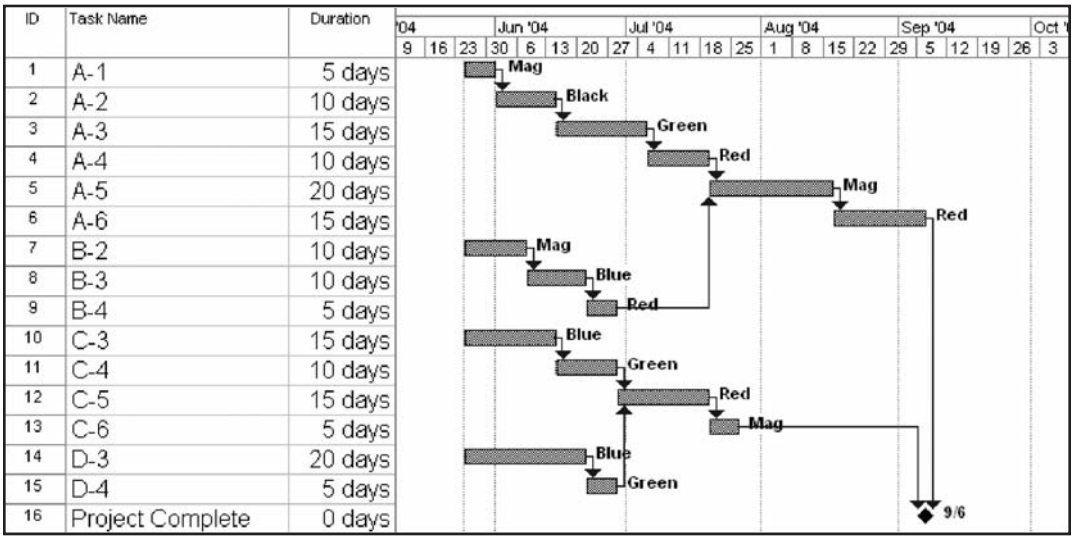


Figure 6.11 Large example Gantt chart illustrates resource contention.

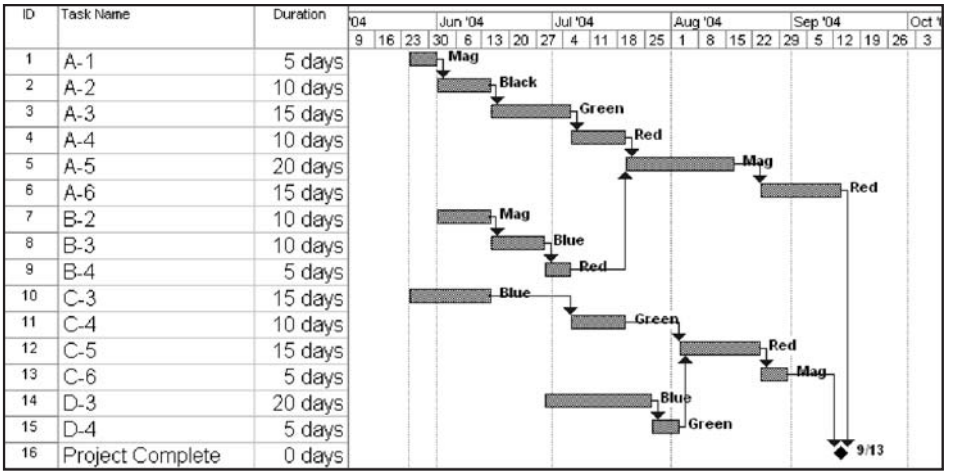


Figure 6.12 Resource contentions leveled.

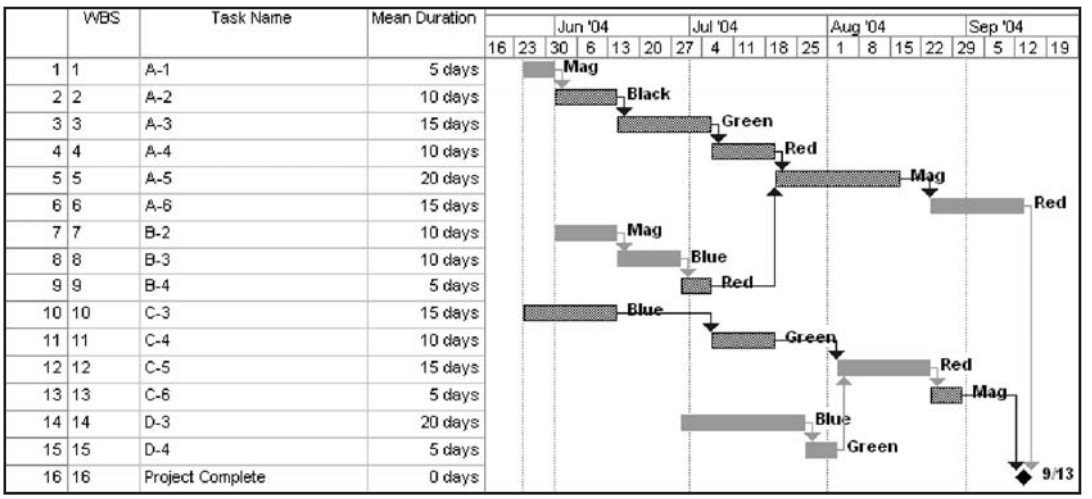


Figure 6.13 Identifying a critical chain.

path several times. You can also see, by tracing backwards from the end of the project, that there are no gaps in the critical chain at this point, even though most logic paths have gaps in them.

Finally, inserting the project and feeding buffers completes the plan. Figure 6.14 illustrates the completed plan. In this case, we sized the project and feeding buffers as 50% of the preceding activity chain. This project requires three feeding buffers.

6.3.3 Large Exercise

Lay out the Figure 6.15 exercise as a critical-chain plan, with all of the appropriate buffers. As above, the first line in each box represents the task number. The second line represents the resource by color. The third line represents the (already-reduced) task time in days. (See the last question in Section 6.11 for the approximate length of the schedule you should have obtained.) Note that a good-enough schedule is within a small part of the project buffer.

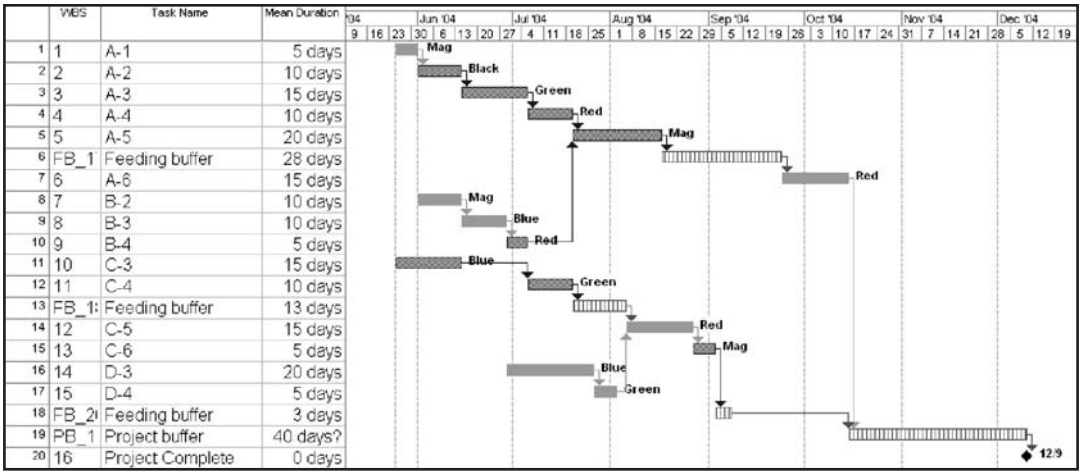


Figure 6.14 Inserting the project and feeding buffers completes the plan.

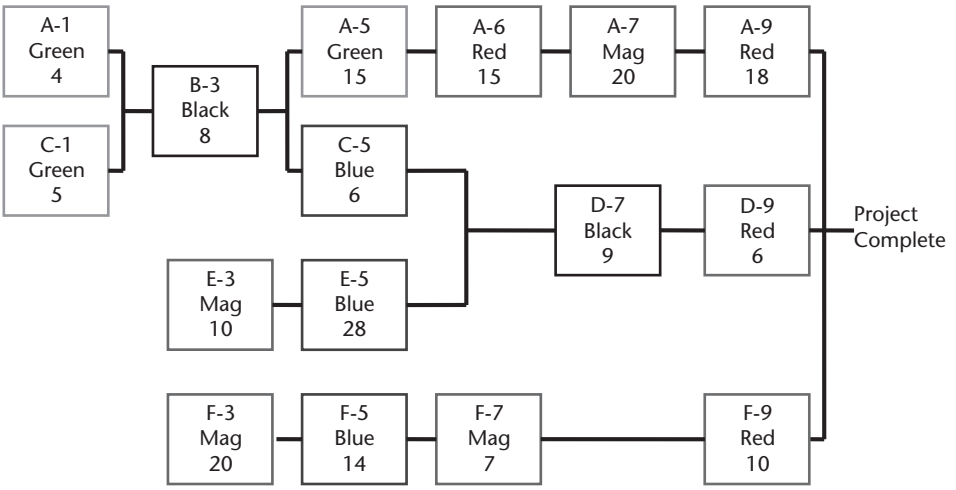


Figure 6.15 Large exercise.

6.4 Buffer and Threshold Sizing

Buffer sizing determines the overall duration of your project and the degree of overall contingency included in the plan. The buffer thresholds for action determine the frequency with which you will act. We usually set buffer thresholds as a percentage of the buffer, so the buffer size influences the actual sensitivity of the buffer triggers.

6.4.1 Statistical Background

Recommendations on buffer sizing use statistics to develop relatively simple rules with a supporting theoretical basis. Dr. Eliyahu Goldratt recommends sizing the project and feeding buffers to one half of the buffered-path task length; that is, do not count the gaps in the chain when sizing buffers. The buffers are there to protect the project from uncertainty in performing the tasks on the chain.

Goldratt's method considers the statistical rule governing the addition of uncertainties that are independent events. The statistical rule says that the uncertainty of the sum of the events is much less than the sum of the uncertainty for each event. This is sensible, as you should expect some variations to be positive and some to be negative. You should consider Goldratt's recommendation in context with his recommendation to reduce activity times in half. Mathematical justification of his recommendation requires several additional assumptions, some of which we highlight below. His recommendation will usually lead to larger buffers than the method described next; this is a reasonable thing to do when beginning to deploy critical chain and a substantial simplification of the process.

Two statistical rules play into sizing buffers. The first is the additive rule for variation. This rule states that for independent events, the variance of the distribution of the sum of the events equals the sum of the variances of the distributions of the individual events. The variance usually represented as sigma, or s^2 is a way to describe the variation of any type of distribution.

The spread in a distribution is proportional to the standard deviation, s^2 or sigma. Thus, the spread of the distribution representing the sum (in our case, the buffer) equals the square root of the sum of the squares of the individual distributions. (Note: Do not worry if you are not a statistics buff and do not follow this. You can do fine with critical chain using Goldratt's simple recommendation or simply by following the procedure we give below. You do not have to know this theory to have it work for you.)

The second statistical rule also works in your favor using critical chain to aggregate uncertainty protection. The central-limit theorem states that the distribution resulting from samples of different distributions tends toward a normal distribution (the central limit) as the sample increases. This means that the uncertainty in performing a number of tasks along a path is a symmetric distribution, even if the distributions of the individual tasks are highly skewed (e.g., have a long tail to the right). This reduces the chances of very long project overruns.

If you make a few assumptions, you can come up with a relatively simple way to use your knowledge of the variation in estimates when sizing the project and feeding buffers. Projects often do not have much information about the actual distribution of the task performance time. (Exceptions might include repetitive projects, such as construction, where extensive estimating data exists.) However, you can usually place bounds on the task time, corresponding to some upper and lower limit of the time it will take. If you assume your estimating method yields about the same meaning for the upper and lower limits on most of the project tasks, you can then say that the difference between this upper and lower limit D is some multiple of the standard deviation. You may not know if it represents two or six standard deviations; you are only assuming that whatever it is, it is about the same for all of the tasks you estimate with the same method. Then, without even having to define the limits precisely, you can size the buffer to protect the whole chain of tasks to the same degree that we were previously protecting each activity. You take the square root of the sum of the squares of the Ds. This result is always less than adding the Ds.

For example, consider a chain of four tasks, each two weeks long. Two weeks is our standard low-risk estimate. One week is our 50–50 estimate. Therefore, D equals one.

The critical-path chain is therefore eight weeks. The critical-chain tasks add up to four weeks. Since *D* equals one, *D* squared also equals one. The sum of *D* squared is then four, and the square root of four is two. Adding the two-week buffer to the four-week task chain gives a project duration estimate of six, compared to eight for the critical path. In this case, the square root of the sum of the *D*-squared method gives the same result as Goldratt's simplified method. This always happens for four equal-length tasks where *D* is half the task duration. That is to say, it doesn't happen very often.

6.4.2 Project and Feeding Buffer Size

CCPM practitioners have come to use a variety of methods to size the project and feeding buffers. You should consider your knowledge of task variation and the maturity of your organization when selecting the method you use. I usually recommend that most organizations use the first method (half the chain) described below, at least initially. The square root of the sum of the squares (SSQ) method tends to appeal to organizations of engineers and scientists. Although it is mathematically more rigorous, experience has not shown it to be better. Method 3 uses an understanding of variation and is where I hope all organizations will trend in the long term.

Method 1: 50% of the Chain This method sums up the task duration along the path and sizes the buffer as half of the total. You should not count gaps in the chain when using this method. For feeding chains that branch upstream, you should consider only the longest path.

This method has two strengths: it is simple, and it usually provides a large-enough buffer. The primary weakness of this method is that it does not allow you to explicitly account for known variation in the tasks. This method also creates relatively long duration buffers on long projects, which some people find difficult to justify.

Method 2: Square Root of the Sum of the Squares (SSQ) The SSQ method uses information on the low-risk and mean duration for each task in the chain. It sizes the buffer as the square root of the sum of the squares of the differences for the tasks along the chain. As with the above method for feeding chains with upstream branches, use only the longest chain or the largest result considering each chain.

The primary advantage of this method is that it allows using known task variation. The primary disadvantage is that it may lead to undersized buffers for long chains. There are two primary reasons for this. The first is underestimation of the actual task variation. The second is that this method assumes that all project variation is stochastic, and reality demonstrates that some is not. Some project variation is a result of bias: things that can make projects take longer, but not make them shorter. The path merge bias is one obvious bias in this category. The statistical basis of this method does not work for variation that includes bias.

Method 3: Bias Plus SSQ This method combines the previous two, using a fixed buffer amount to account for variation, summed with the SSQ method to account for common-cause variation. The fixed amount will usually be significantly less than 50%

of the chain. Table 6.1 provides some guidelines to use when determining the fixed amount of buffer to include in the project, feeding time, and cost buffers. You do not necessarily need to sum the corrections. You should use project experience to adjust the bias correction.

Additional Guidelines The following guidelines help assure an effective buffer:

1. Seek to have at least ten activities on the critical chain. *Reason: The more activities in the critical chain, the more effective the sum of the squares and central-limit theorem.*
2. Do not allow any one activity to be more than 20% of the critical chain. *Reason: The uncertainty of one large activity will dominate the chain, leaving little possibility for the other tasks in the chain to make up overruns in the dominant task.*
3. Do not allow the project buffer to be less than 25% of the critical chain. *Reason: Chains with many tasks of uniform length may calculate a relatively small buffer, providing inadequate protection.*

6.4.3 Buffer Trigger Points

Whenever your buffer penetration exceeds the red trigger, you should take action to recover buffer. The reason is that you may not have enough buffer left for the remaining project tasks to ensure that the project finishes with less than 100% of project buffer penetration (i.e., on time).

CCPM sets buffer trigger points to plan for management control action and to initiate the action. Both of these trigger points must be set to minimize false signals and to ensure needed action. There is little negative impact from too low a threshold for the yellow trigger point. You may do significant damage to your project, however, if you set the action (red) trigger too low and take unnecessary control actions. Project changes, which include control actions, are very likely to cause confusion and delay the project.

You should track buffer penetration over time. Most CCPM software has the ability to track buffer penetration. All of the software implements buffer criteria that vary linearly over the planned length of the critical chain, commonly called a fever chart (see Figure 6.16). The idea is that the yellow-to-red transition should cause you to take action to recover buffer when the remaining buffer is insufficient for the remaining tasks. This would require a dynamic recalculation of the buffer as the project proceeds. The CCPM software replaces this dynamic calculation with a linear approximation and in some cases allows the user to adjust the thresholds. Table 6.2 lists typical thresholds.

Table 6.1 Guidelines to Size the Fixed Portion of Buffers

Cause of Bias	*Range of Time Buffer*	*Range of Cost Buffer*
Omissions	Some, not to exceed cost impact	5%–10%
Path merging (more than 5 parallel paths)	Up to 20%	None
Errors	5%–20%	5%–20%
Special-cause variation	0%–30%	0%–30%
Failure to report necessary rework	0%–20%	Covered by errors

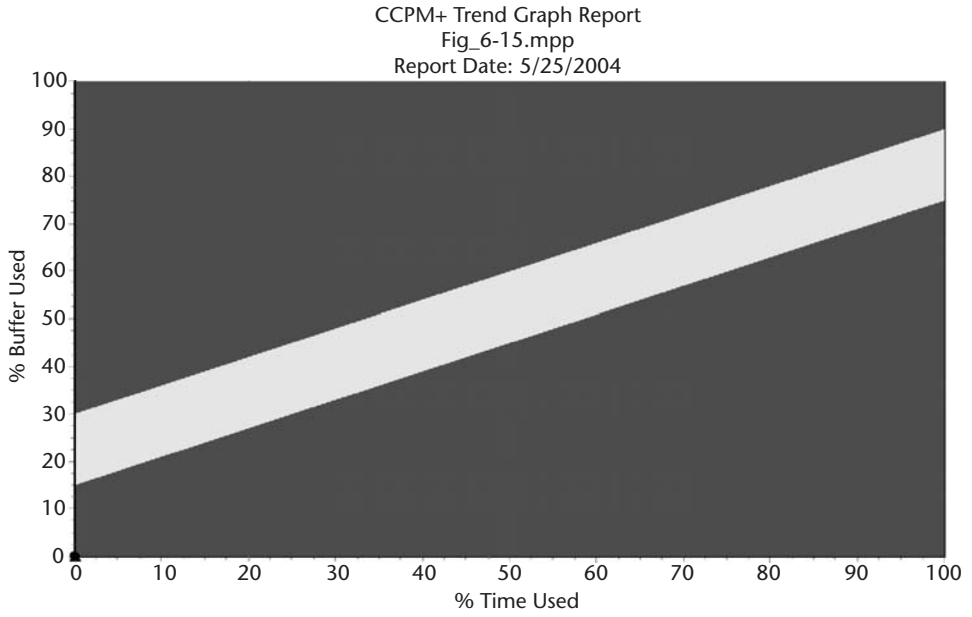


Figure 6.16 Project buffer tracking with a fever chart.

Table 6.2 Buffer Threshold Settings

	Buffer Penetration (%)	
Critical Chain Complete (%)	*Green-to-Yellow Transition (%)*	*Yellow-to-Red Transition (%)*
0	15	30
100	75	90

If you are tracking the buffer over time, you may wish to institute some of the additional control-chart triggers, such as four points in a row tending toward the trigger point. Do not make the trigger logic too complex.

Set the buffer triggers for feeding buffers at the same percentage of buffer penetration as you do for the project buffer. However, you should prioritize tasks for work considering only their impact on the project buffer.

6.4.4 Resource Buffers

Goldratt recommends using a resource buffer to help resources along the critical chain anticipate when they might receive a task to work on. Notification may become necessary in CCPM because we no longer associate dates with every task, and resources sometimes need to plan ahead (e.g., not plan a vacation when a critical-chain task is likely to arrive). Of course, the dates we used to give had zero probability anyway, but never mind.

I find the filtering capability of project software can perform the functional task Goldratt wished to achieve with the resource buffer, at least for internal resources. Most scheduling software updates the ongoing schedule with actual start dates and duration as you status the project. Filtering the project plan for a particular resource gives you the latest best estimate of when all tasks will start.

The Concerto Software autmoatically prioritizes all tasks based on tehir impact on the projet buffer. This greatly simplifies focusing resources on the rights task. You can create similar resource-filtered lists to perform this functions with other schedule software.

For subcontractors, you may need to provide delivery dates. Be sure that the dates are early enough to not delay the project, should it be ahead of the baseline plan. You may consider making the resource buffer for a subcontractor a financial incentive to ensure a lead-time. Since profits are a small percentage of revenue, you are often able to greatly increase delivery reliability by doubling suppliers' profits if they deliver on time. This should only cost you a small percentage of the subcontract.

6.5 Cost Buffer Sizing

Use a cost buffer if your business is project-cost sensitive. Organizations using throughput accounting and internal projects (e.g., internally funded R&D) may not require a cost buffer.

Sizing the cost buffer requires considering a number of factors. First, you should budget for the use of the schedule buffers. While start delays will not directly translate to cost, additional activity-duration times used by people working to complete the activity will increase cost. You should include at least 50% of time buffers into the cost buffer at an appropriate cost rate related to the chain that they protect. Alternatively, you could sum the amounts removed using one of the buffer-sizing methods you used for time, such as bias plus SSQ (i.e., the cost buffer equals the square root of the sum of the squares of the cost removed from each project activity, plus a fixed amount for bias). Note that this method is subject to the same considerations that applied when using the sum of the squares to size time buffers (e.g., for many nearly equal cost activities, this method may yield a much smaller buffer).

Second, you must consider the unique aspects of each project that affect your ability to estimate accurately. For example, if you are estimating unique materials or materials subject to wide price variations, you should consider this when sizing the cost buffer.

Finally, you should take advantage of using an aggregated cost buffer. This substantially reduces the total cost buffer requirement. It also reduces the tendency to "use it or lose it," which sets in if you include cost contingency in each activity. As with schedules, because of human behavior, projects do include cost contingency. The only question is if you have a readily identified, aggregated contingency under the control of the project manager or hidden contingency at the discretion of each task performer.

Never attempt to operate with a cost buffer of less than 10% of the estimated project cost. The reason is that there is always some bias in project cost estimates. You can always forget some things and underestimate others. Project reviews will usually remove any additional "unneeded" items in the cost estimate and make sure that individual cost estimates are not unrealistically high.

The buffer size for fluctuations should consider that you will rarely get the advantage of work-package underruns (estimated at their average durations), but you should consider the statistical combination of the positive variances. If you have

a dominant work package in terms of total project cost and uncertainty, the uncertainty in that work package should size the statistical part of your cost buffer. If your work packages are similar in size, and you have several of them, you can use the square root of the sum of the squares to size the statistical contribution to the cost buffer.

If your customer is dissatisfied with the size of the cost buffer, you might consider rolling-wave planning. This method phases the plan, with a higher level of detail and lower level of uncertainty associated with near-term, better-known tasks and less detail and more uncertainty for later phases of the project. The rolling-wave method adds detail to the plan periodically as it is better defined.

If your organization uses cost-and-schedule-control reporting or uses project plans to sum up organizational resource demands, you can add the cost buffer into the project plan. We recommend you put it all into the project buffer. If you use other means for global resource planning, you can put it into the buffer as a leveled fixed cost. If you use the individual project plans to project resource demand, you must put in a resource distribution representative of the aggregated project. For example, divide up the people resources to represent the same percentage in the buffer as they do in the plan.

6.6 Methods to Create the Plan

People have successfully used a variety of methods to make and control critical-chain plans. Initial critical-chain projects all used some type of manual method. Keep in mind that we are cautioning against putting too many tasks in a critical-chain plan (i.e., a critical-chain plan should have no more than a few hundred activities. Less than a hundred is preferred.)

6.6.1 Manual

The simplest and most commonly used method to manually create work packages is to use the PERT chart format and sticky notes. This may be all you need for very small projects. The procedure is as follows:

1. Fill out a sticky note for each task, containing the task ID, title, duration (reduced), and controlling resources. (You may wish to use color-coding to identify the task-duration controlling resource.) Indicate the tasks that provide needed input on the left side of the note.
2. Lay the notes out on a board or table according the task logic and following the rough time logic (this is called a time-phased PERT or a time-phased logic diagram).
3. Remove resource contentions.
4. Identify the critical chain.
5. Add sticky notes for the project and feeding buffers.
6. Size the feeding buffers.
7. Calculate the critical chain using a “forward pass.” Starting with the initial task, write the start times on the lower left corner of the note and the

completion time (start time plus duration) on the lower right corner of the note.

8. Calculate the feeding paths using a "backward pass" from where they enter the critical chain.
9. Remove any remaining resource contention and revise calculations.

This process is not difficult for projects with fewer than about twenty tasks. It gets harder after that, as you need a lot of real estate to lay the project out.

You may refine the method by cutting out colored paper bars to represent each task. The length of the bar represents task length, and the bar color represents the controlling resource. This simplifies the resource-contention steps and subsequent calculations. It obviously requires a little more up-front preparation. Large projects with more than 500 tasks have used this method successfully. A magnetic scheduling board is another tool to implement this same idea.

You may extend this method to use computer software such as PowerPoint or Excel.

6.6.2 Critical-Path Software

You may use critical-path software to plan and manage critical-chain projects. Most software packages have sufficient options to support you in leveling the resources and using late start on the feeding chains. You always start from the same place: with a project logic containing the reduced task times and resource requirements. You should ensure (when necessary) that you have selected the appropriate options to maintain the fixed task duration that you input and that you have selected options to late-start each path. Sometimes, you can do this globally. Other times, you can put constraints on the first task on each path that causes the downstream tasks all to late-start. (You need to experiment and understand what your software does to these options or constraints during resource leveling.)

Most critical-path software provides options for the algorithm to perform resource leveling. You may experiment with these. The critical-chain method does not depend on the algorithm you use. It simply requires that the final plan remove all resource contention within the single project. Usually, you can do resource leveling manually if you wish and view the final resource allocations by task.

After initially leveling resources, you must identify the critical chain. I suggest you add links to the plan to cause the resource leveling to stay in place. You can then remove other constraints that your software may have added to implement resource leveling (e.g., some software adds fixed-task-start-date constraints to implement resource leveling.) If you do add logic connections, you should then be able to calculate the schedule and have the critical path equal the critical chain.

Assure that the critical chain you identify really is the constraint of your project. Sometimes an inadvertent logic connection results in tasks on the critical chain that cannot or should not determine the duration of your project. (I call this a mathematical critical path or chain.) Adjust logic or task duration to cause the critical chain to be a legitimate constraint to your project. (Note again that there may be two nearly-equal-length paths vying for the critical chain. I suggest you choose the one that you feel has higher uncertainty or that makes most use of a potentially capacity-constrained resource.)

Also, assure that the distribution of tasks on the critical chain will provide effective immunity from variation in any one task. I suggest two simple guidelines for this:

1. Ensure that the critical chain comprises at least ten tasks (unless your project is very small).
2. Ensure that no single critical-chain task comprises more than about 20% or your critical chain or more than 50% of your project buffer.

Next, you need to add the feeding and project buffers. You add these as tasks without resource requirements. Remember to tie in the feeding buffers as predecessors to the critical-chain task at the point they join the critical chain. You must then recheck your resource leveling and make any final adjustments. (Adding the feeding buffers usually requires redoing some amount of resource leveling.)

6.6.3 Critical-Chain Software

Critical-chain software automates most or all of the process described above. Several software packages are currently available. The most widely available software products currently provide add-ins to Microsoft Project or use Microsoft Project within their framework.

6.7 External Constraints

Projects may have external constraints. These factors may influence the project lead-time and are not under the control of the project team. Regulations, inspections, and permissions often fall into this category. External constraints may be internal to the company; for instance, another division might have to provide an essential component.

The TOC five focusing steps provide a method to deal with these constraints. First, you have to identify them as constraints, or as potential constraints, and deal with them accordingly. If they are only potential constraints, you can deal with them under project-risk management. If you feel that the potential for them to become constraints is large, you may want to ensure that they are on the critical chain.

The second step is to exploit the constraint. In the case of regulations and permits, this usually requires providing a high assurance that all submissions to the regulators meet their needs completely. This may require additional resources up front. You should consider, however, that any delay in the project critical chain should be valued for the burn rate of the entire project or the expected daily return upon completion of the project. You may elect to hire experts in the particular area to help ensure success. There may be portions of the project that can be exempted from the constraint.

The third focusing step subordinates everything else to the constraint. This may require doing additional scope or investing additional management time to ensure good working relationships with any people or agencies that may become external constraints.

It is improbable that you will elect to elevate an external constraint.

6.8 Reducing Planned Time (a.k.a. Dictated End Dates)

Project managers are often asked to accelerate schedules. With CCPM, there may be a tendency to look at the juicy project buffer and suggest that reducing the buffer is a "painless" way of reducing the planned project lead-time. Reducing the project buffer has no impact on project execution time. Reducing the project buffer only reduces the chances that you will meet your promised lead-time and causes excessive buffer triggers. Excessive buffer triggers damage project performance. Do not cut the project buffer!

6.8.1 Acceleration without Cost Impact (Exploit and Subordinate to the Constraint)

Several sensible methods can reduce project lead-time. Preferred options do not increase cost. Two primary options are to get additional resources where resolving contentions caused the lead-time to be increased and to look inside the tasks for batching opportunities (i.e. reducing batch sizes).

You may only need an additional resource for a short time to make a significant improvement in the overall project lead-time. If there is a way to obtain the additional resource, this method can reduce the overall project lead-time at no additional cost since you had to perform the tasks for the same individual durations; that is, you did not change the task work (person-days). You may reduce the project buffer if this change reduces the length of the critical chain.

Batching occurs when a task includes more than one physical output. For example, a task may include making a number of certain parts used in the final assembly. The parts may be identical or different. Parts are not limited to hardware. They might include different technical products, such as drawings, parts lists, or reports. The parts might even include different people, such as hiring people to staff one shift at a time.

The successor task may be able to start when the first of the predecessor outputs is available. In this case, you can break up the task into smaller pieces to better show the real workflow. Your plan can also show this type of relationship as a task start-to-start dependence with a lag. Alternatively, you can show it as a finish-to-finish task logic. Whichever way you choose to present it in the plan, your management process should ensure that performers understand and focus relay-racer performance on each individual task output. They must keep the sequence to realize the assumptions made in your plan.

If batching involves a significant number of parts, you may wish to invoke a supplemental method to track and control the parts through the repetitive process. Your critical-chain plan would show this process as a single activity (e.g., "process 37 parts"). One effective method uses the "line-of-balance" method, combining features of operational process control with project management. The line-of-balance method plans the time for each part to traverse the process flow, creating an expected number of parts through each step at a given time (the line of balance). Tracking compares the actual parts through each process step to the line of balance.

6.8.2 Acceleration with Increased Raw Material Cost (Elevate the Constraint)

You can also reduce project time by exercising higher-cost alternatives. For example, you can use overtime or hire additional temporary resources (which usually cost

more). You may be able to purchase components with a higher cost but shorter lead-time. You may be able to use higher premiums for early subcontract delivery.

The TOC suggests that considerations of increased cost should compare the additional operating expense to the impact on project throughput. The throughput of project acceleration (per day) is the value of the whole project (per day). Compare the cost of increased raw material cost to the throughput increase from the acceleration. If the throughput increase exceeds the cost increase, you should elevate the constraint. The throughput increase usually greatly exceeds the cost. In two projects I recently worked with, one day's acceleration could mean $10 million and $18 million dollars, respectively. Any payment made to accelerate was immediately worthwhile.

6.9 Enterprise Wide Resource Planning

You may use enterprise wide scheduling tools effectively to identify to resource managers the anticipated window and duration of tasks they have to support. Resource managers must buy-in to understand that the dates are not meaningful; we focus on *windows* of performance and task duration *estimates*. The resource managers can then assess if their long-term aggregate demand requires more or fewer resources. They can allocate resources based on the criteria given above.

6.10 Frequently Asked Planning Questions

Sometimes "abnormal" things seem to happen when following this procedure. In addition, questions arise. The following are answers to some frequently asked questions.

1. After we add the feeding buffers, is is a problem if noncritical chains start earlier than the critical chain?

   This can happen and should not be a cause for concern. Start the project with the noncritical chain. Some CCPM software allows you to toggle a task such as this onto the critical chain. Do this if it makes you feel better.
2. When we add the feeding buffer to a noncritical chain with a critical-chain task as a logical predecessor, it "pushes" the critical-chain task back, creating a gap in the critical chain.

   First, you should examine your plan logic and resource loading to try to eliminate such gaps. For other cases, consider where the noncritical chain feeds the critical chain and the relative variability of the two chains. Remember, we are subordinating everything else to the critical chain. In general, gaps in the critical chain should only happen because of a company constraint resource. Gaps in the plan do not mean that you should have a gap in performance.

   If your project contains a number of parallel paths with roughly the same duration, you might consider leaving out the feeding buffers in the parallel paths and increasing your project buffer instead. If you do this, you

must pay extra attention to tracking progress on these parallel paths, as you will not have the feeding buffer measurement to help.

3. Why don't we connect the other chains by their resource and path dependencies?

   Attempts to add that level of detail do not improve project performance. Variation will occur, so controlling every dependent chain is not possible. The buffer-sizing approach, feeding buffers, and, most importantly, using buffer management to guide resources on which task to work on next provide the necessary, sufficient buffering and control.

4. Our schedules have thousands of tasks. How can we plan the project without an effective computer program?

   I recommend that you confine the top-level project schedules to a few hundred tasks, at most, and use work packages or subprojects where more detail is required. CCPM projects have been successfully completed with more than 30,000 tasks in the entire project.

   Be sure you are using your schedule to manage the handoffs of tasks and not as a part list or checklist. Some computer software allows you to add task notes or checklists and links to other information for detail. You can also use the WBS to link to task detail. Experience demonstrates that the more detailed tasks in the schedule, the more often you have to revise the schedule and the greater the probability of error. This leads to long turn-around times for schedule updates and the loss of control.

5. We have tasks in our project plan over which we have no control. What should we do?

   Regulator or client review of project outputs often creates this situation. You can control what you give them and when you give it to them, but you cannot (directly) control their work processes. In this case, working with your stakeholders, as described in Chapter 1, will provide great benefit. You can influence how long their review takes and limit potential rework by going to the effort necessary to ensure you understand their requirements and produce a quality product for their review. If you are a significant part of their workload, you can help them focus by staggering your submissions to help them avoid multitasking. Other unique situations demand unique solutions. You should use the five-step focusing process in those instances.

6. Our (management/client) has specific intermediate milestones they want us to schedule a date for and meet. What do we do?

   This may occur for a number of reasons, including coordination of work with other parts of a larger project. We know of cases where contracts tie project payment to satisfactory completion of these milestones.

   If satisfying these milestones creates throughput for your company, we recommend planning milestone accomplishment as a project of its own. You can then use the multiproject method to link the projects.

   If satisfying the milestones is simply a tracking tool, we suggest you first try to convince your management or client that buffer reports are actually a better tool. Failing that, we suggest you put the milestones at the end of a feeding buffer, or use a special milestone buffer. All completion dates must

be preceded by a buffer. (If they do not fit on the critical chain, they are not the right milestones, and we again suggest option one.)

7. Our client does not want our result early because we are a subassembly to their project, and they do not want to have to store our input. What do we do?

   Use the critical-chain process to schedule the start of your activity chains to satisfy the client needs. Usually, this will mean you can delay some activity starts.

8. Our projects cause us to discover new work as we go. How do we handle this?

   When your projects inevitably lead to new work (e.g., maintenance activities that first must diagnose what is wrong), you should include your best estimate for the "discovery" work in your initial plan (i.e., not your worst case). You should also structure your plan to reduce risk by determining the additional work as soon as possible in your plan. You can add tasks to your project as you go, as long as this does not cause your project buffer to exceed 100%. Once your project buffer penetration exceeds 100%, and you cannot come up with a realistic buffer plan, it is time to replan your project and resize the buffers.

9. Our critical-chain software is placing tasks that are not critical before tasks that we know to be more critical. What should we do?

   The critical-chain software honors the task-predecessor (technical) logic you gave it and builds a resource-feasible plan. If you have not told the plan that one task is more important than another, it will place tasks to produce a reasonably short schedule. This may not be the optimum schedule from a technical or risk-management perspective.

   Some software allows you to prioritize tasks in your network and then adjust the resource leveling to ensure that the resources go to these tasks first. Some CCPM software does not enable this capability. If you have this issue and are not using software with this capability, you have two choices. You can change your model, or you can change your execution. You can change the model with various strategies, such as using multiple projects or additional task links. You can change the execution by using judgment to override the schedule task sequence when possible (i.e., when predecessors are complete for two tasks competing for a resource that is coming available, assign the resource to the one you know to be higher priority).

10. What are the answers to the two exercises presented in this chapter?

    There are multiple satisfactory solutions to each exercise. If your results come within about 15% of the project buffer to the total lead-time below, they are "good enough."

Project	*Critical-Chain Length*	*Project Buffer*	*Total Project Lead-time*
Small exercise	50	25	82
Large exercise	107	47	154

6.11 Key Points

This chapter has described how to create a critical-chain plan for a single project. The steps up through creating a logic diagram with low-risk duration estimates do not change from the reference PMBOK™ approach. The critical-chain steps are as follows:

1. You should start planning a CCPM project, as you would any project, by determining the scope of work and network of tasks necessary to deliver the project scope. You can use a variety of tools to help you accomplish this.
2. You must resource-load and -level CCPM project plans before you determine the critical chain.
3. Start with estimates of duration as you usually make them and then allocate some of that estimate to the task and some to the buffer.
   - Size project and feeding buffers using a method appropriate to the maturity of your organization. In most cases, it is best to start with using 50% of the preceding chain of tasks.
4. Project- and feeding-buffer trigger points determine the frequency of control action. Decide on them as part of your plan.
5. The (optional) cost buffer provides aggregated cost protection in the same way that the project buffer protects the schedule.
6. The TOC five focusing steps (identify, exploit, subordinate, elevate, prevent inertia) provide a framework for resolving environment- and project-specific issues.

Constructing a critical-chain plan is a relatively small addition to the work necessary to construct an effective critical-path plan. It may be less work and create a more useful plan if you reduce the level of detail in your plan. Even when it requires work you have not done before (e.g., resource-loading your plan), the benefit far exceeds the investment.

CHAPTER 7

Developing the Multiproject Critical-Chain Plan

7.1 Identify the Multiproject Constraint

The critical chain is the constraint for a single project. What is the constraint of an enterprise that performs multiple projects? How do you put the critical chains of multiple projects together in a way that identifies the constraint of the enterprise to produce projects that meet the three necessary conditions, and do it in a way that allows focus on increasing the project throughput of the enterprise? What is it that constrains the enterprise from completing more projects or completing the existing projects more quickly?

Consider a reference environment that most people are familiar with: mowing a lawn. Consider the amount of grass cut as the counterpart to completed projects. What happens when the grass is too long or when you try to push the lawn mower too fast? It bogs down and often stalls.

The same thing happens when you push too many projects into a multiproject environment without considering the capability of the constraint to perform the projects. If you push too many projects into the system, it will bog down and stall. People will work very hard, but projects will take a long time to complete (the engine is stalled much of the time), and a lot of management effort goes into restarting the engine and cleaning out the debris. It will seem like there are never enough of the key resources necessary to complete the projects.

With the lawn mower, you use the feedback from the system to adjust the rate of processing. You listen and slow down the lawn mower as the engine begins to slow down. Alternatively, you raise the cutting height to match the processing rate to the feed of the work. Normally, you do not run right out and purchase a new lawn mower or rebuild the one you have. The organization counterpart to buying a new lawn mower is hiring more staff, and efforts such as reengineering are similar to rebuilding the one that you have. A simple adjustment (raising the cut height of your lawn mower) gets you back in business. Likewise, a simple adjustment to how your organization plans and releases projects can have a huge impact on the bottom-line result.

Figure 7.1 illustrates an example critical-path multiproject scenario. Common project activities share the same resources. Using conventional low-risk activity estimates and considering three-project multitasking, each activity duration is 90 days.

Considering this and the project in Figure 7.1, assuming these projects are all the same, the resources have to be divided between the three projects, even if you

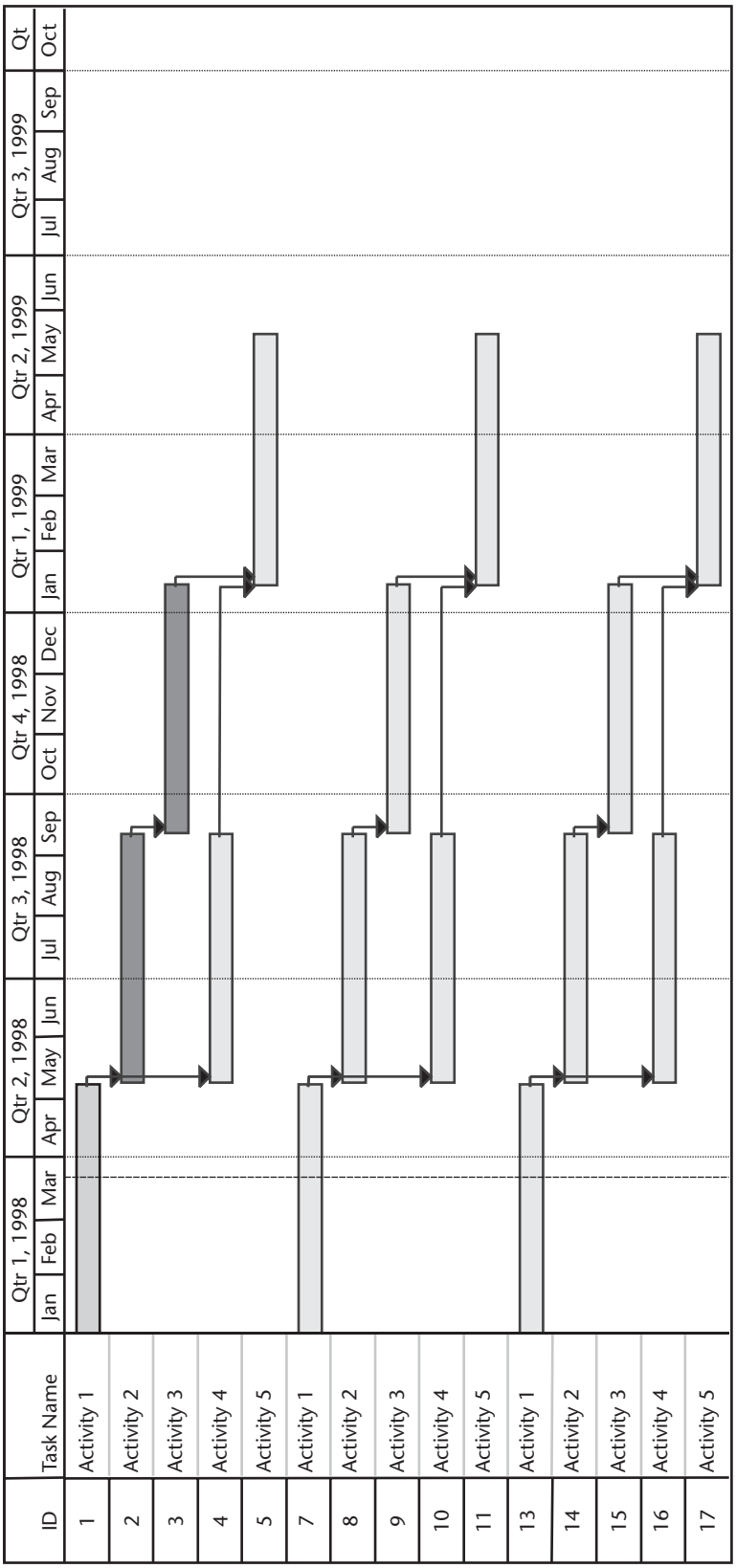


Figure 7.1 Three projects in a multiproject environment.

have only one resource of each type. Thus, either the project plans assume that the resources will multitask across projects (i.e., are working only a fraction of the time), or the projects are not going to complete on time due to the reality of multitasking. In the case of identical projects, it is easy to understand that one resource is likely to be overloaded more than the others are. That resource is the capacity constraint of the system (i.e., the whole organization cannot complete projects any faster than the most loaded resource can complete its project tasks).

Some companies do check resource availability across all projects. They then argue to increase resources (buy more lawn mowers). This is moving to the elevate stage of the TOC before completing the identify, exploit, and subordinate steps. This is a very expensive strategy.

To improve throughput you have to first *identify* the company capacity-constraint resource. This is most often a certain type of person but may be a physical, or even a policy, constraint. The company constraint resource becomes the drum for scheduling multiple projects. This terminology comes from Dr. Eliyahu Goldratt's production methodology, where the drum sets the beat for the entire factory. Here, the drum sets the beat for all of the company's projects. Think of the drummer on a galleon. What happens if even one rower gets out of beat?

The project system becomes a "pull" system because the drum schedule determines the sequencing of projects. You might recall this major precept of the Lean approach to manufacturing. The system should pull projects forward in time if the drum completes project work early. The system should delay subsequent projects when the drum is late. For this reason, projects in a multiproject environment require additional buffers to protect the drum, to ensure that the system never starves the capacity constraint for work. You must also schedule projects to ensure that they are ready to use the drum resource, should it become available early.

Figure 7.2 illustrates the CCPM method. Compared to the previous critical-path case, CCPM reduces each activity time to 15 days to eliminate the three times multitasking (i.e., resources splitting their time between tasks, one on each project) and to use 50% probable duration estimates. The CCPM approach identifies the resource supplying activities 2 and 3 as the capacity-constraint resource. CCPM exploits the resource by synchronizing the projects using this resource as the drum. CCPM subordinates to this resource by adding capacity buffers between the projects. The capacity buffers ensure that the capacity-constraint resource is available for the subsequent project.

Figure 7.2 shows the CCPM plan completing the three projects (including the project buffer) near the end of August 1998. It shows the first two projects completing even earlier. Contrast this to the critical-chain multiproject plans of Figure 7.1, all of which are scheduled to complete in May of 1999. Based on what you have learned for single projects, you can expect the CCPM projects to be early. Based on experience with critical-path projects, we can expect them to be late even for these extended schedules.

Also note that synchronizing the projects this way reduces resource contention for all resources, not just the drum resource. This happens in the example because the projects are identical. While most multiproject environments do not have identical projects, synchronizing projects to the drum usually eliminates some, if not all, resource contention. Resource manager's prioritizing resource assignment to

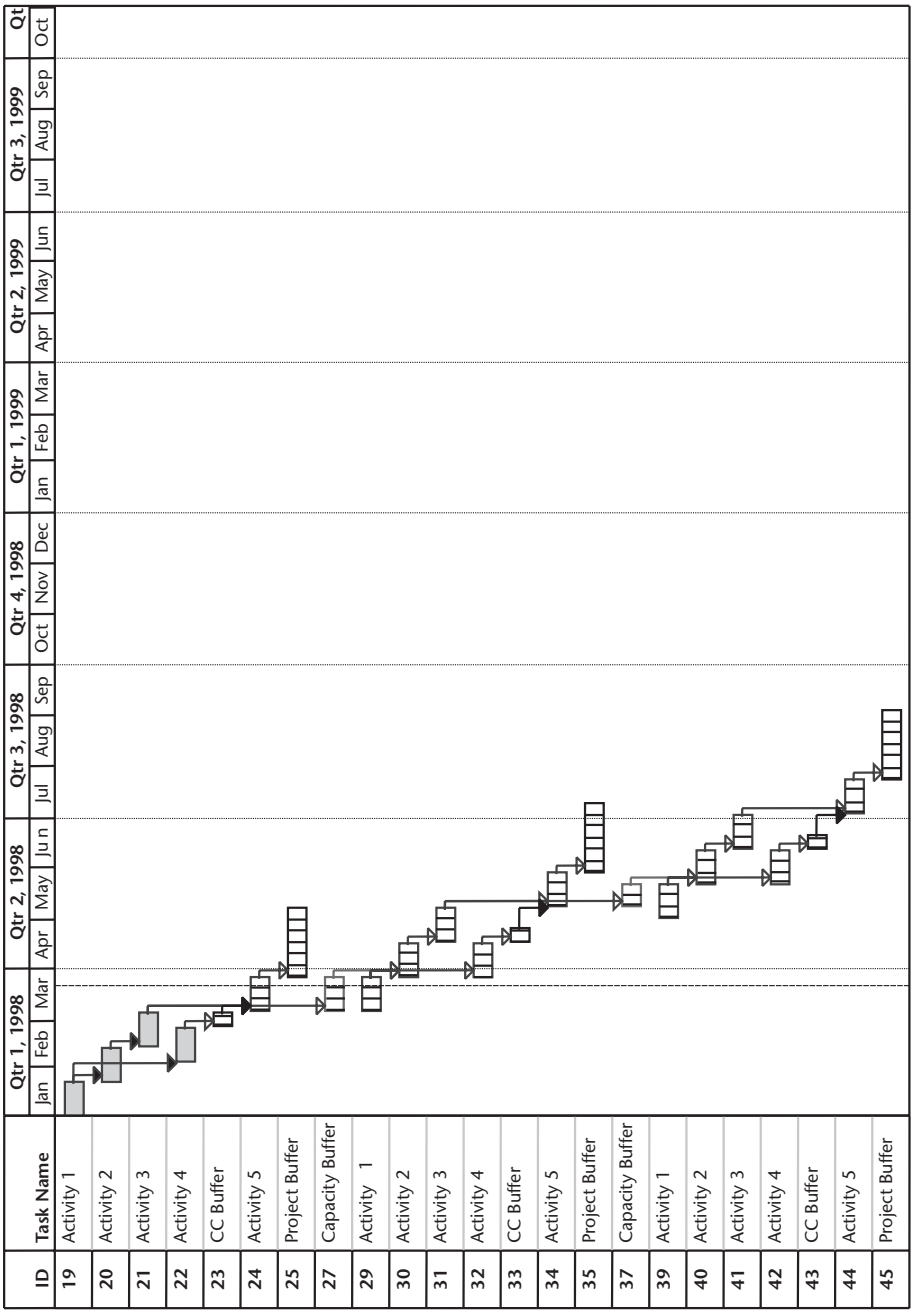


Figure 7.2 CCPM multiproject plan reduces project duration and increases project throughput.

work on tasks according to the penetration of project buffers resolves remaining resource contentions.

This is a major simplification compared to attempts to micromanage a whole organization. These attempts never work. I hope that by now you understand the reason why this is a hopeless exercise: all of the activity durations are estimates. None of the activities should take the exact amount of time planned. Any schedule produced for all resources across all projects is a fiction. It is only one possibility out of millions of possible combinations of project status and resource availability. Instead, the critical-chain process provides a dynamic process to allocate resources: buffer management. CCPM allows for this variation with the project and feeding buffers within each project. This process also includes the ability to absorb the natural variation in the buffers. It is a real-world control system.

The TOC leads to an understanding that *all resources other than the constraint must have excess capacity*. Those upstream of the constraint resource must have excess capacity to ensure that the constraint resource is never starved for work, as this would waste its capacity. In a project, this means we have to buffer to ensure that we provide the constraint resource with the input it needs. Resources downstream of the constraint must have capacity more than the constraint in order to deal with fluctuations in their own output and that of resources between themselves and the constraint resource. They must ensure that they always deliver the constraint-resource-processing rate to the completion of the project(s). This is the concern of the project, not of the constraint resource.

While projects theoretically can have resource demands in any order, there tends to be a similarity in the order within a company, based on the type of projects they operate. For example, many projects will have a design phase, procurement phase, construction phase, and initial operation phase. Thus, the sequence of demands on resources tends to be similar, although the usage may vary substantially from project to project. The general idea carried over from manufacturing is that the further a resource is from the constraint resource in the plan sequence, the more excess capacity and/or the larger buffer it needs to not influence the overall lead-time.

7.2 Exploit the Multiproject Constraint

The constraint resource becomes the drum for the company projects (like the drummer on the ancient galleons setting the pace for the rowers.) Therefore, the procedure to exploit this resource is as follows:

1. Identify the company constraint resource.
    1.1. The company constraint resource should be the resource that determines the greatest amount of critical-chain duration on your projects. It will usually be apparent as the resource that is frequently in short supply and often called on to use overtime. If several resources exhibit the same behavior, select one based on the unique contribution of your company (e.g, if you are an engineering company, your drum resource should be some type of engineer, not an administrative

resource). Otherwise, select the one usually in demand nearest to the beginning of a project.

2. Exploit the company constraint resource.
   2.1. Prepare the critical-chain schedule for each project independently.
   2.2. Determine the project priority for access to the constraint resource.
   2.3. Create the constraint-resource multiproject schedule, or the drum schedule. Collect the constraint demands for each of the projects and resolve contentions between the projects to maximize company throughput (i.e., to complete the most projects early).
3. Subordinate the individual project schedules.
   3.1. Schedule each of the individual projects to start based on the constraint-resource schedule.
   3.2. Designate the critical chain as the chain from the first use of the constraint resource to the end of the project.
   3.2. Insert capacity-constraint (cc) buffers between the individual project schedules, ahead of the scheduled use of the constraint resource. This protects the drum (constraint) schedule by ensuring we have the input ready for it.
   3.3. If inserting the cc buffers influences the constraint-resource schedule, resolve contentions.
   3.4. Insert drum buffers in each project to ensure that the constraint resource will not be starved for work. Place them immediately preceding the use of the constraint resource in the project.
4. Elevate the capacity of the constraint resource.
5. Go back to step one, and do not let inertia become the constraint.

The following sections describe the features of this process.

7.3 Multiproject Critical-Chain Features

7.3.1 Project Priority

You must prioritize all ongoing projects before creating the drum schedule. This prioritizing serves one purpose: to set the priority for use of the drum resource. Your method for setting the priority may consider a number of factors, but the primary factor from the TOC is to maximize the company throughput per use of the constraint. If you have a direct measure of project throughput, you can actually use this ratio to set the priority, dividing the project throughput (usually in dollars) by the drum resource demand (usually in person-hours or person-days).

Legitimate reasons for other considerations in setting the project priority should consider the company goal. For example, it may be advantageous to give higher priority to your best customers, considering your need to make money in the future.

7.3.2 Select the Drum Resource

The drum resource must be shared across all projects you consider part of the multiproject environment. This is the definition of a multiproject environment. Larger

companies may have several independent project groupings that share resources within the group but not across groups. In this case, you should have multiple drums.

Resources often appear as constraints. The company capacity constraint sometimes may seem to "float." The basic TOC makes it unlikely that there is in fact more than one constraint (unless you have an unstable system)! Statistical fluctuations can make temporary capacity constraints. For example, suppose a number of projects happen to demand a particular resource at one time, exceeding the resource capability. This is a statistical occurrence. We should expect it to happen. It does not make this resource a company-capacity constraint. It does mean our project plan and control system have to handle it, even if only through the individual buffers we have already added. There is also some flexibility in resource supply. On occasion, we can use overtime or ask people to defer time off. We can segment the work to ensure that we are properly exploiting the potential constraint. We can subordinate other work that does not produce immediate throughput.

However, many companies will have a chronic resource constraint. The department is always on overtime or always seems to run late. It has presumably been permitted to occupy this position because of some policy or for another reason that prohibits providing enough of the resource to meet all demands. If two or more resources seem to contend for this honor, pick the resource demanded nearest the beginning of projects. That leaves you the option to change your mind later if necessary. We can call this the capacity-constraint resource because it influences overall company performance. There must be a reason that we cannot easily increase the supply of this resource. This resource is the company bottleneck and, therefore, must become the drum for all of the projects.

Since the purpose of selecting the drum resource is to stagger the start of the projects and avoid overloading the system, it usually does not matter much if you select the wrong resource as the drum. You will still get some degree of project staggering. As long as you choose a relatively highly loaded resource that you cannot easily elevate, you are likely to get a large benefit. Project performance will help you focus on the correct drum resource over time. It is far better to get on with the drum schedule with the wrong resource than to continue to operate the old way while agonizing over the actual drum resource.

Many criteria have been proposed to identify the drum resource. With project plans, you do have the total resource demand, and you should know your total resource on hand. You could select the drum by the highest ratio of demand to available staff. You should only use this method if you have some reason to believe both of these numbers for all projects. Goldratt does not recommend this method for production because he claims your data is never very good. This may also be true for projects. If you use this method, you should ensure that the resource selected is not easily elevated, for example by hiring contractors or temporary staff.

In order to achieve the maximum effect of staggering the projects, the drum resource should be the resource that controls the largest amount of critical-chain time on your projects. This resource may vary from project to project. If, like many companies, your projects tend to follow a repetitive pattern (e.g., from engineering to construction to operation), you may find one resource that dominates critical-chain time. Selecting this resource makes it most likely that you will remove resource contention for all of the other resources in the project.

Avoid Assigning Resources by Individual Name Many companies choose to identify resources by individual names. They feel that the resources are so highly specialized that they cannot do otherwise. If this is true, you have no other option. I will observe, however, that your company is at high risk if your total multiproject throughput is controlled by one or more individuals who, if they leave or get sick, will bring all of your projects to a halt. You should consider this situation as part of your project-risk-management approach.

The preferred approach is to assign resources by type in your plans and then have the resource manager assign specific individuals as the task comes up to be performed. The definition of a resource type, then, must be that any person with that designation could do the tasks assigned to that resource type. The primary advantage to assigning resources by type is that the larger the resource pool, the more opportunity you have to dynamically assign resources to projects as the activities demand. This applies to all resources, not just the drum resource. You may, when the task allows it, further accelerate tasks by assigning more than one resource of the type to the task.

7.3.3 The Drum Schedule (a.k.a. Pipelining the Projects)

The drum schedule is the plan for allocating the drum resource across all projects. The manager who has responsibility for the drum resource usually manages it. The drum schedule is the primary determinant of the system capability to process projects. It sets the start date for each project.

Some like to call the process of creating and maintaining the drum schedule pipelining. The idea is that you are maximizing the flow of projects through the project pipeline, one after another. This does not require that one project complete before you start another, but rather that you enter projects into the pipeline to best utilize system capacity.

The drum manager (which some call the master scheduler) needs the drum-resource demands for each project and each project's priority to create the drum schedule. The individual critical-chain project plans determine the duration, earliest time, and relative times for each of the drum-using activities in each of the projects. Figure 7.3 illustrates the drum-resource demand from three projects, positioned from highest priority on the bottom to the lowest priority at the top. The drum schedule must fit in all three projects while not exceeding the capacity of the drum resource. For the example shown, there are two units of the drum resource.

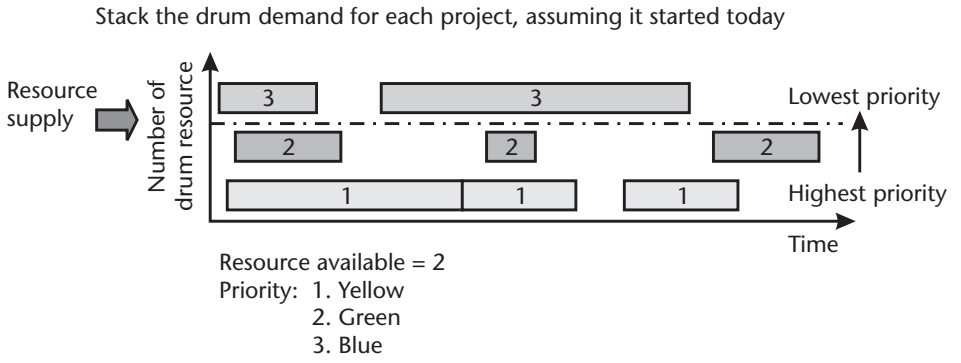


Figure 7.3 Three-project drum-resource demand, assuming all three projects are to start today.

Note that the drum-resource use cannot be scheduled earlier than shown in Figure 7.3. This is because other activities on the projects have to feed the drum-resource using activities. These are the earliest times that the projects can use the drum resource.

The method is to push the lower-priority projects later in time until they fall in under the resource demand. This creates the drum schedule. Note that when scheduling the drum, the task duration taken from the individual project schedules is the average duration. Since you will want a low risk of not having the drum resource available, you must allow time in the drum schedule for longer-than-average actual duration. You accomplish this by including the capacity-constraint buffer in the drum schedule. Figure 7.4 illustrates the resulting drum schedule.

7.3.4 The Capacity-Constraint Buffer

The capacity-constraint buffer assures that the constraint resource is available when needed by the project. Conceptually, you place it between the use of the constraint resource in the prior project and the first use of the resource in the project you are scheduling. It does not take lead-time out of the project you are scheduling, but it defines the start date for the resource-using activity.

The previous paragraph uses the word *conceptually* because the actual process of sizing the capacity-constraint buffer with multiple projects can be much more complex than visualized by delaying one or two tasks relative to another set of tasks in another project. Think more in terms of filling a bucket. The bucket represents your resource capacity. You will usually want to put the "big rocks" in first. These projects have a firm deadline and perhaps are contracted with delivery penalties. You must do these projects as soon as possible. The big rocks do not fill all the space in the bucket. There is space between the rocks. Your bucket still has some room. So next, pour in the gravel, the smaller projects, in accordance with their priority. After that, there is still some more room to pour in the sand, the nonproject work. That still leaves room for you to add water to the bucket, the crises that do not really involve project work.

Some project software allows you to specify the time bucket for resource leveling (e.g., weekly or monthly). It will allow overallocations and not try to level resources, as long as the average demand for the time bucket is within the average

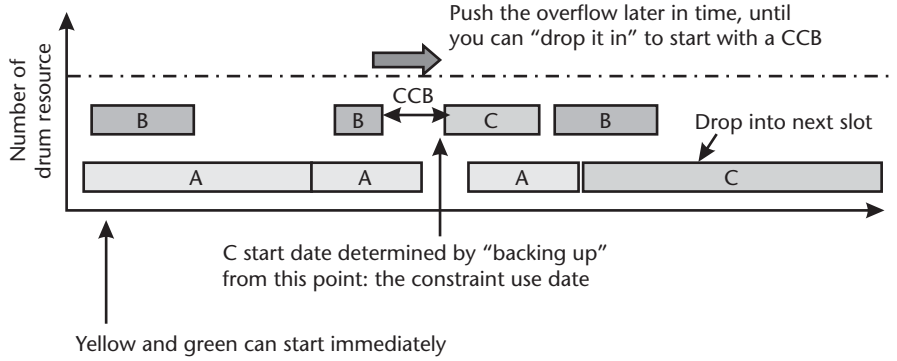


Figure 7.4 The drum schedule accommodates all project demands, including capacity-constraint buffers.

supply of the resource. This fits well with CCPM, as we know that those apparent overlaps are not real; they are an artifact of presenting a reality with variation using a deterministic drawing method.

You should consider queuing theory and your resource-leveling approach when sizing the capacity-constraint buffer. Queuing theory suggests that the capacity-constraint buffer should be at least 25% of the capacity-constraint-resource capability. Otherwise, your projects will slow down to a crawl.

Everyone is familiar with queues. We wait in queues at the supermarket, at the bank, and, one after another, in airports. Some of us even sometimes wait in queues to go to the bathroom. We all know that queues can form very rapidly, and they can dissipate rapidly when extra servers are applied.

I ask project-management groups, "Suppose that the average time to process each person through a queue exactly equals the average arrival rate of people to be served by the queue. How long will the line be?" Most people answer that there will not be a line, or that the line will average one person waiting to be served. Unfortunately, this is an excellent example of the human mind's inability to intuitively understand variation. For this case, over time, the line approaches an infinite length. Of course, it takes an infinite time to get that long, but it can grow surprisingly rapidly and, once there, will not dissipate until the server capacity is increased or the arrival rate decreases. That may be a reason stores close the doors at night.

Figure 7.5 illustrates the classic queuing curve for one line and one server. It plots the length of the line versus the ratio of the average arrival rate to the average processing rate. The curve for wait time has the same shape. A value of $x = 1$ means

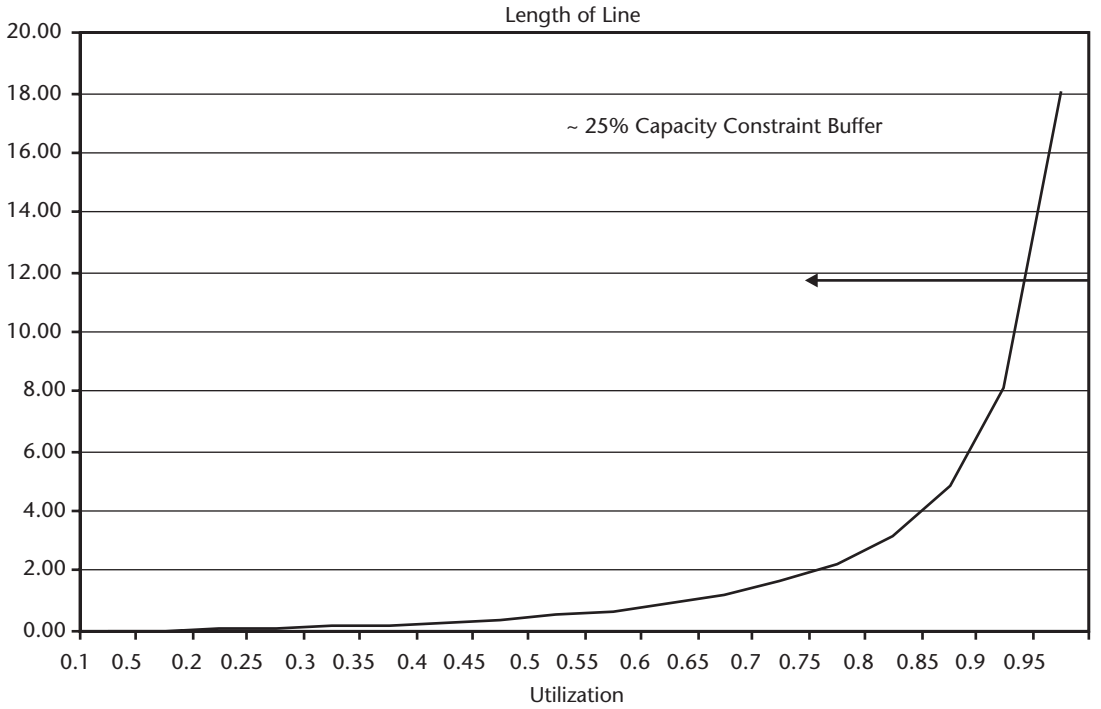


Figure 7.5 The queuing model predicts an infinite line when the average arrival rate approaches the average processing rate. [For project resources, utilization = (average arrival rate of tasks)/(average duration of tasks).]

that the average arrival rate equals the average processing rate. Note that the line is infinitely long at that point and rises very rapidly as x approaches that level. The queuing model has certain statistical assumptions that underlie it, but the overall behavior is quite robust. The line begins to grow very rapidly as the ratio gets beyond about 0.7, or 70%, average utilization of the resource, which corresponds to a 30% capacity-constraint buffer.

You might appreciate the following to help understand this surprising result. Consider that you are working at 90% capacity and are sick for a day. It will take you nine days to catch up because you only have 10% excess capacity available each day to catch up. Now suppose you are working at 95% capacity. It will take more than twice as long to catch up because you have half as much time each day to make up the loss, and you have lost a little more. At 99%, it takes 99 days to catch up. At 100%, you can never catch up.

The reality is that people make up for a lack of capacity buffer by making excess capacity. They will work additional hours, paid or not. They will find innovative ways to move the work on. They may cut corners so that the backlog does not get too large. They may send on incomplete work. They may send on lower-quality work. Although some amount of overtime for a limited duration (i.e., a couple of weeks) can be beneficial when focused by CCPM, research consistently demonstrates that extended overtime leads to a total throughput that goes back to or declines lower than preovertime levels.

If you do not want your projects to wait an infinite amount of time for the drum resource, you should use a capacity-constraint buffer in the range of 25% to 30%.

7.3.5 The Drum Buffer

Goldratt envisioned the drum buffer to ensure that the drum resource has input to work on when it is needed in the project. In this respect, the drum buffer is a feeding buffer. Do not confuse the drum buffer with the capacity-constraint buffer. The idea is to place the drum buffer in the project schedule immediately prior to the activity using the drum resource. (The capacity-constraint buffer is not visible in individual project plans; CCPM uses it for project staggering.) The drum buffer can directly affect the project's scheduled start date and lead-time if it is on the critical chain. The drum buffer is usually on the critical chain, but it is not necessary that it be.

I generally recommend not worrying about the drum buffer in your initial implementation of CCPM. You may find that you do not need them at all. Using the CCPM project-execution tools and tracking progress daily usually render them unnecessary.

If you do use a drum buffer, you should size it as if it were a feeding buffer, using the upstream activity path as the chain length.

Some have suggested sizing the drum buffer using an arbitrary lead-time, such as 14 days. I do not understand the basis for this recommendation, other than that it stems from a concern that management may have a tendency to exploit the drum resource in a negative sense. Since a properly sized drum buffer (previous paragraph) will usually have the activity input ready before the drum resource is available, this may tend to put pressure on the drum resource to multitask or hastily complete the prior task. Either of these behaviors will have negative consequences. Avoid them.

7.3.6 Project Schedules

Once you have the drum schedule, you create the individual project schedules by aligning the start of the project to match up with the drum-using activity. In other words, you work backwards from the drum-using activity to schedule the start of the project. Since you had to have the project critical-chain schedule with a "time now" start date to create the drum schedule, this amounts to delaying the start of some project by the amount you had to delay the drum-resource-using activity to fit it into the drum schedule, plus the drum buffer. You then schedule the rest of the project downstream from the drum-using activities.

Note that you are not leveling all resources across all projects. Instead, you plan each project assuming the most efficient level of resources for that project and then stagger the project starts to level the drum resource only across all projects. This minimizes the cycle time of each project, while maximizing the throughput of the organization. Leveling all resources across all projects will slow everything down: it is like trying to cut the deep grass all at once.

7.4 Another View of a Multiproject Constraint

Multiple projects can present special situations. Whenever you have a special situation, you should go back to the basic definition of the TOC and review the five focusing steps. Taking this approach has led some recent large projects I have been engaged with to view the system constraint differently. In two of these cases, there was a multiproject program, which is a multiproject environment more or less on its own (i.e., the multiproject environment has a dedicated group of resources, and the problem is how to get the overall program completed as soon as possible). One case consisted of a large number of complex repairs to an existing structure, and the other case consisted of fabricating a number of complex pieces of equipment necessary as part of an even larger, complex system.

I am indebted to Mark Woeppel, a TOC consultant from Plano, Texas, for the following analogy. In these cases, you can think of the overall product of the program as a race car. The program completes when the racecar completes the race. This a long race, with drivers that switch out. The drivers of the racecar are the resources, moving the overall program along. They are the ones working on the actual equipment. A number of technical limits prohibit having various combinations of resources working on the equipment at the same time. For example, you cannot weld while spray-painting, and only so many people can fit into confined spaces.

Many resources support the drivers. When the racecar comes into the pit, every resource must have the tools and supplies it needs to optimize the performance of the drivers and move the car along as rapidly as possible.

When you have a unique constraint like the race car, the planning work begins with identifying the critical chain of each project, then working to reduce the overall duration for each project. Next, you can pipeline (sequence) the projects to the most constraining resource (drum). Nevertheless, you are not done. The next set of tasks to exploit the constraint require you to act like members of the pit crew and prepare detailed plans to maximize effectiveness when the car pulls into the pit.

7.5 Introducing New Projects

New projects can arrive in a multiproject environment at any time. You will have a list of prioritized projects and a drum schedule, and you will know the status of all of the ongoing projects. You have to fit the new project into the system.

The only way to schedule a new project is through the drum schedule. To do this, management must first decide the new project's priority. It may be of the lowest priority if management prefers the first-in, first-out priority method, or it may be of higher priority than some of the ongoing projects. For example, the new project may be for a very important customer, and therefore, management may want to give it higher priority than in-house projects.

You then must prepare the critical-chain schedule for the new project to determine when (in relative time) it will demand use of the drum resource. You can then fit this resource demand into the proper sequence in the drum schedule. The drum schedule determines the start time for the project by backing up from the time the drum resource will be available for the new project.

If the new project is placed at a higher priority than some of the ongoing projects, the schedule of the ongoing projects will change. This may lead to an interruption of work. You should use common sense when interrupting project work (e.g., you should not interrupt nearly completed tasks or tasks that do not have immediate demand for the resource from another project). Management should consider the potential impact of these interruptions when placing a new project at higher priority than an ongoing project.

Figure 7.6 illustrates the introduction of a higher priority-project into a drum schedule. You first put it into the schedule, assuming the project will start right away, but above the next lower-priority project as illustrated in Figure 7.6. You put projects of lower priority than the new project above the new project. Then, you fit in the drum use as best you can as illustrated in Figure 7.7. This may lead to suspending some ongoing projects. If you do suspend ongoing projects, you should do so wisely (e.g., do not stop nearly complete tasks without completing the task result).

Always keep in mind that the worst possible priority decision is not to make a priority decision, to encourage everyone to do his or her best. This inevitably causes bad multitasking and the worst performance on all of the projects.

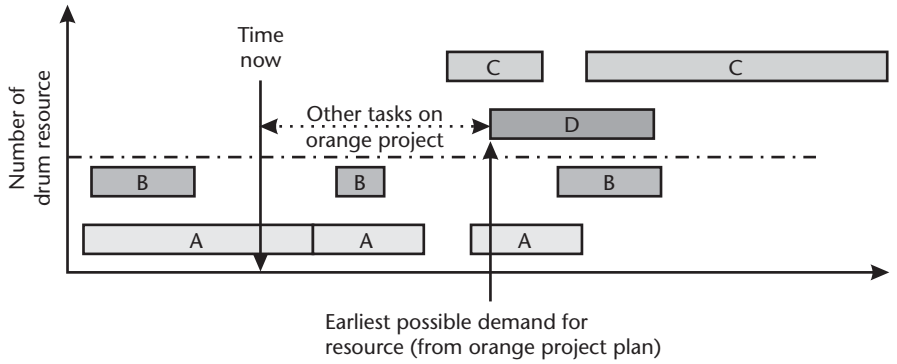


Figure 7.6 A new project is added to the drum demand and judged by management to be higher in priority than an ongoing project.

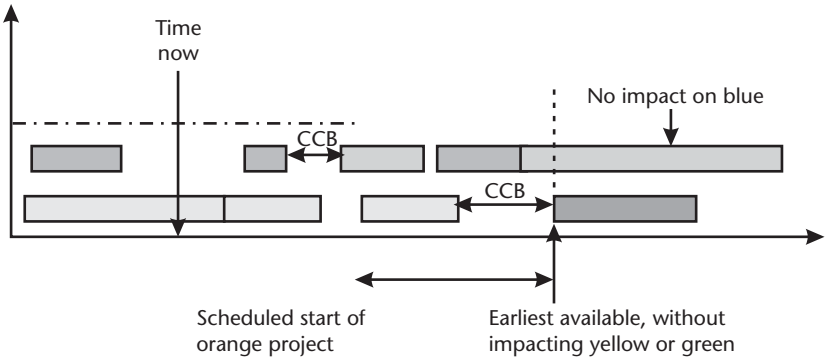


Figure 7.7 Resolving the drum demand sets the schedule for the drum resource in each project, including the new project.

7.6 Frequently Asked Multiproject Questions

1. Our projects change over time. How do we identify the drum?
   It is a good idea to check the drum resource from time to time, as it can change. When it does, it is time to reanalyze the projects. Often, even though the drum resource changes, the amount of delay necessary for projects to work effectively does not change that much.
2. Our projects tend to all go through the same phases in parallel. Thus, the constraint changes over the course of the year. How do we identify the drum?
   This frequent behavior is the result of a mind-set suggesting all projects should start at the same time (e.g., the start of the fiscal year or the start of the calendar year). My good friend Scott Button, a CCPM consultant, compares this to a snake swallowing a pig: the lump moves down the project system over time. The project system needs to overcome this policy constraint first.
3. Can we have multiple drum resources?
   An organization can have multiple drum resources if groups of projects share essentially nonoverlapping resource pools. In that case, you can identify a priority list for each group of projects and create a separate drum resource schedule for each group.
4. How can we make management adhere to a priority list?
   If management does not adhere to a priority list, the multiproject system will not work. It is a simple choice, really. Behave to double throughput, or do not. Once they see the results, many management teams are able to do much better at this than many thought. After all, when the system makes more money, people's jobs are protected, and often they make more money too.

7.7 Summary

The critical chain for a single project is usually not the constraint for an enterprise performing multiple projects. It is necessary to identify the multiproject constraint

and go through the focusing steps to adapt the CCPM process to firms with multiple projects. When you identify the multiproject constraint and use it to schedule projects, it is the drum for your organization. The following list summarizes the key points for CCPM multiproject planning.

- The drum resource is the constraint in a multiple project environment.
- Management must select the drum resource and prioritize all projects for access to it.
- The capacity-constraint buffer ensures that the drum resource is available when needed. The capacity-constraint buffer should be in the range of 25% to 30% of the constraint-resource capacity.
- The drum buffer assures that the drum resource is not starved for input.
- Individual project critical-chain plans operate to the project-chain start times developed from the drum schedule, including the capacity-constraint buffer and drum buffer.
- Management must introduce new projects to the system through the drum schedule by first assigning their priority relative to ongoing projects, then scheduling the drum-using activities.

Practical applications of CCPM have demonstrated the greatest gains in multiproject enterprises. The reason for this is that those environments usually require everyone to multitask much of the time. Elimination of much of the bad multitasking has the greatest impact on overall enterprise project throughput.

CHAPTER 8

Measuring and Controlling to the Plan

The chapter presents key CCPM measurements and considers two types of measures: operational measures and performance measures. The task and project managers have most need for operational measures, while senior management and resource managers have more need for performance measures. Operational measures are tactical, while performance measures are strategic.

In the following list of what effective measures must do, Dr. Joseph Juran [1] identified the first six items. He seems to have missed the seventh. Section 4.3 describes the background for exploiting the plan using buffer management and notes that Dr. Eliyahu Goldratt defined information as "the answer to the question asked"; this is covered by the eighth item in the list. Number 9 is both a TOC and a Lean precept [2]. Effective measures must do the following:

1. Provide an agreed basis for decision making;
2. Be understandable;
3. Apply broadly;
4. Be susceptible to uniform interpretation;
5. Be economic to apply;
6. Be compatible with existing design of sensors;
7. Provide early warning of the need to act;
8. Deliver information to the person who must act;
9. Be simple.

The Lean precepts of focusing on flow and using visual controls were also important considerations when designing an effective measurement-and-control system for CCPM. The system must eliminate the *Muda* (waste) of complexity.

The goal of all projects in all organizations must relate to the company goal. For a profit-making company, this goal is to make money now and in the future. As noted in Chapter 1, for projects that do not have a specific date deadline, performing a project to meet the customer's needs for the budgeted cost on or before the committed delivery date will support that goal. For projects with a firm date (Olympic-stadium-type projects), meeting the date is the goal. Project measures must provide information so that people can make local decisions that favorably impact the global project goal.

Please consider as we go on how the CCPM measurement-and-control system satisfies those requirements.

8.1 Project Roles

This chapter addresses measurement and control to the plan for four different roles: the task manager, the project manager, the resource manager, and senior management.

8.1.1 Task Manager Role

The task manager's job is to maintain project flow. The task manager needs to know which task to work on next. The trick is to work on the tasks that will move the project toward completion as soon as possible. W. Herroelen, R. Leus, and E. Demeulemeester [3] criticized CCPM for not "rescheduling" the project dynamically: "Opportunities for speeding the remaining part of the ongoing project may be exploited by rearranging the schedule" (p. 57). They did not understand that CCPM does not schedule task dates at all, and by performing task assignments dynamically, it achieves the same end as continuous rescheduling, without the continuous rescheduling: accelerated project completion. Although CCPM uses resource leveling to determine the overall duration necessary to complete the project, it a mistake of subtle deterministic thinking to think that the resulting task dates actually mark when tasks will take place. Tasks will start when the predecessor task is complete and the resource is available, and they will complete as soon as possible. Herroelen, Leus, and Demeulemeester's paper actually made a good case for following the CCPM approach.

The first rule of CCPM implements relay-racer task performance: Once resources start a task, they should complete it as soon as possible. As resources complete tasks, task managers should put available resources to work on the next task that is (1) available to start (i.e., the predecessor has completed), and (2) causing the most project-buffer penetration. This is true within a project and across multiple projects. That task can be on or off the critical chain. In a multiproject environment, project-buffer penetration can be higher on a lower-priority project. In that case, the resource should work on the task on the lower-priority project before working on the higher-priority project. Project priority is implicit in the projects' start and end dates once they have been pipelined by the CCPM multiproject approach.

You can greatly facilitate this decision by providing the task manager with a prioritized list of tasks in the order in which they should start. You can automate this process with some non-CCPM software. For example, a recent client, using Primavera, instituted a script to generate the prioritized list. The list put the started tasks at the top and then prioritized, using a float calculation from a working schedule in which the feeding-buffer durations had been reduced to zero. The list sorted from least float (including most negative) to greatest float and had a column to identify that the predecessor task was done.

The CCPM+ software [4] takes a graphical approach, showing the tasks that are ready to work in priority order (Figure 8.1). The Concerto software automates the process for multiple projects, providing the task manager a prioritized list of tasks designed to facilitate rapid task-statusing and communication (Figure 8.2).

The Concerto software [5] takes a strongly user-focused perspective to provide near real-time reporting of project information. The Web-enabled software allows anyone with permission to access it anytime, anywhere. Task managers can input

task statuses at the end of their shifts in just a few minutes. Most projects that use it run the buffer analysis daily; more intense, multishift projects sometimes run the analysis twice a day. A system administrator authorizes different user roles, all accessing the same database. This ensures that all stakeholders operate from the same data. Alternative analyses and views of the data support the different decisions necessary for the different roles.

During project execution, the task manager plays the key role in making the whole CCPM system work: reporting task start and completion and estimating remaining duration (RDU) for in-progress tasks. The estimate of RDU drives the determination of buffer penetration and, therefore, can impact everyone's task priority. Task managers must be competent and committed to making realistic estimates, and they must be accountable to ensure that the RDU estimates are input into the schedule software in time for project meetings. Task managers are trained to take whatever action necessary to ensure that the information is put in on time, no matter where they are or what they are doing.

For delivery of the task result, the buck stops with the task manager. There are no excuses: no matter what happens to the resources or what occurs on the task, the task manager is accountable to deliver the task result in the shortest time possible. If resources are not available or are ineffective, the task manager must take whatever action necessary to resolve that problem. If at any time the task managers feels unable to resolve a problem to move the task to completion, he or she must immediately engage the help of the project manager and, if appropriate, resource managers.

8.1.2 Project Manager Role

The project manager controls the project value stream. During execution, the project managers' primary question is, when should I take action to recover buffer? The

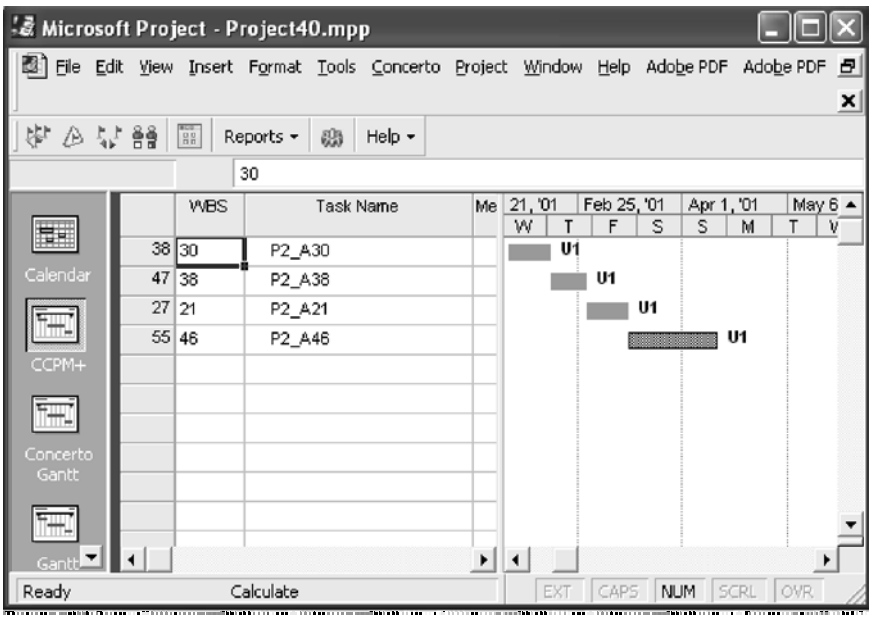


Figure 8.1 CCPM+ Software shows a prioritized, filtered view for task managers to decide which task to work on next. The first three tasks are on the critical chain.

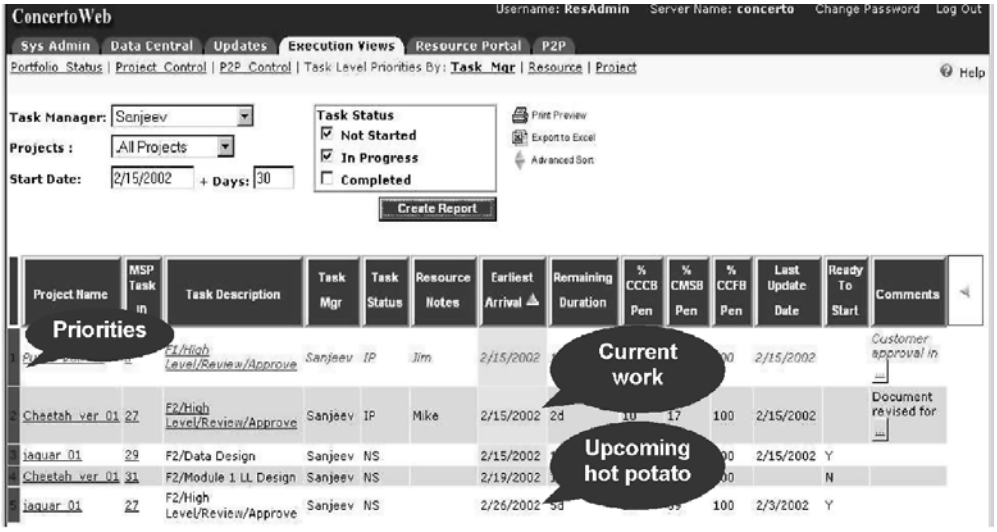


Figure 8.2 The Concerto software uses a Task Manager view to support all of the task manager functions across multiple projects.

project manager also must ensure smooth handoffs between the task managers to maintain flow and aid the project team in communication and problem solving. The project manager also controls project changes.

In addition to buffer-report-driven decisions, project managers also address how to deliver technical quality on time and for or under the estimated cost. Project-level operational decisions include the following:

- Disposition of material that is not up to specifications (this includes, for R&D projects, not getting the hoped-for result);
- Requests for additional time or money to complete activities;
- Requests to add scope (some day, some project may even have a request to reduce scope!);
- Unanticipated resource conflicts;
- Late activities that may threaten the delivery date;
- Unanticipated external influences (e.g., accidents, weather, new regulations) and unfulfilled assumptions (e.g., soil conditions dictate a need to put in pilings before construction);
- Recovery from mistakes.

Project managers should monitor the project buffer and each feeding buffer at the appropriate time intervals for the project, usually daily but at least weekly. For buffer management to be fully useful, the buffer monitoring time must be at least as frequent as the shortest task duration. If the buffers are negative (i.e., the latest activity on the chain is early relative to the schedule date) or less than the red buffer-penetration criteria, the project manager should not act on the project. If the buffer penetrates between the yellow and red thresholds, the project manager should watch

closely and plan actions (create buffer-recovery plans; see Section 8.5) for the buffer-penetrating chains to accelerate the current or future tasks and recover the buffer. If the buffer penetrates by more than the red criteria, the project manager should implement the planned buffer-recovery action. This process provides a unique, anticipatory project-management tool with clear decision criteria.

Note that managing the feeding buffers helps protect the overall schedule from delays in merging paths, including paths that merge at the project buffer. Action criterion for the feeding buffers can be the same as for the project buffer, although frequently feeding buffers penetrate very rapidly, or not at all.

Buffer reporting relies on realistic estimates of how many days are left to complete on a task, or RDU. There is often a tendency to report a project as "on schedule" until the due date arrives. With the critical-chain measurement system, that amounts to subtracting the days worked on the task from the total duration estimate. Project managers should question estimates that are repeatedly on schedule. A useful aid to estimating is to ask people, particularly on the critical chain or on feeding chains with significant buffer penetration, to explain the basis for their RDU estimate.

Project managers should use plotted trends of buffer utilization. This provides a measure and trend of the rate of buffer consumption by plotting the amount of buffer consumed versus the amount of critical chain accomplished as illustrated by Figure 8.3. The buffer measure then acts as a control chart, and the project manager can use similar action rules; that is, any penetration of the red zone requires action. Four points trending successively toward the red zone require action. Trending is especially important if your processes to produce project tasks are not in statistical control. Walter Shewhart [6] notes that the trend information is even more important in such cases.

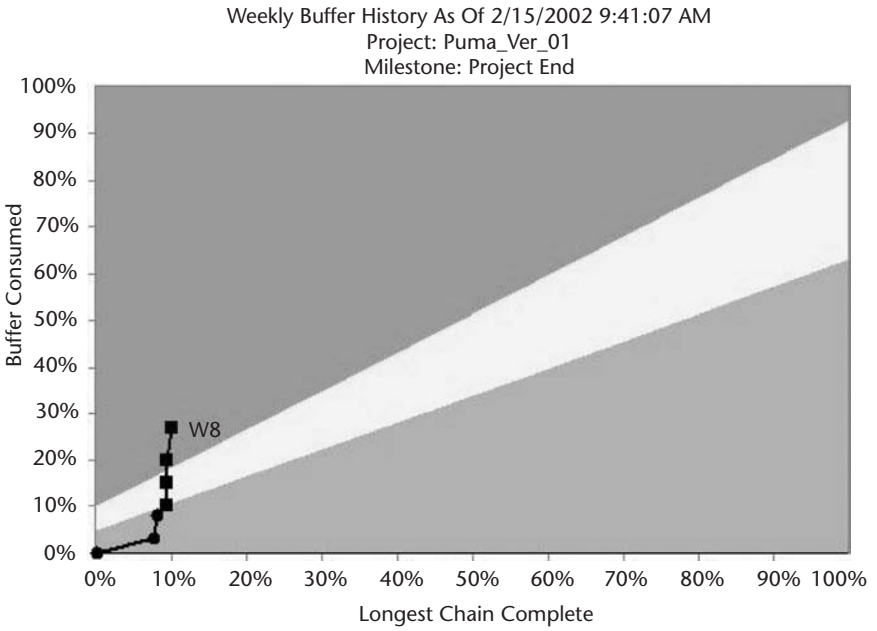


Figure 8.3 Measuring the trend of buffer penetration improves the early-warning aspect of buffer management.

Figure 8.4 illustrates the Concerto Software's project chain view. This view shows the tasks in order that are causing project buffer penetration, and the amount of buffer penetration. This tells the project manager where to focus buffer recovery.

8.1.3 Resource Manager Role

The resource manager provides qualified resources to do the project work. Section 8.1.1 covers the portion of a resource manager's role when he or she is acting as a task manager. Resource managers also fulfill a strategic role, ensuring that appropriately skilled resources are available as needed on an enterprise and project level. The manager of the drum resource may also act as the master scheduler for the organization, developing and maintaining the drum schedule.

Resource managers can use outputs of schedule programs to support performing their role. Microsoft Project provides a resource graph, giving the resource manager a view of the long-term demands on a particular resource. Most scheduling software also supports filtering for the tasks that use a particular resource. The filter works with a variety of views, including the Gantt view. This lets the resource manager see the tasks that are coming.

The Concerto software provides a unique resource portal for the resource manager. Figure 8.5 shows the ratio of demand to supply for each resource over a user-selectable time frame. The first column lists the resource. The two numbers in the right columns illustrate the demand and availability of the resource, with the demand in parenthesis. The bar chart in the second column illustrates the overall resource load over the selected time horizon as a percentage of the available resources. The tool allows drilling down to the actual tasks that create the demand. The resource manager also has a view similar to the Task Manager view, showing a prioritized listing of all of the tasks from multiple projects placing demands on that resource.

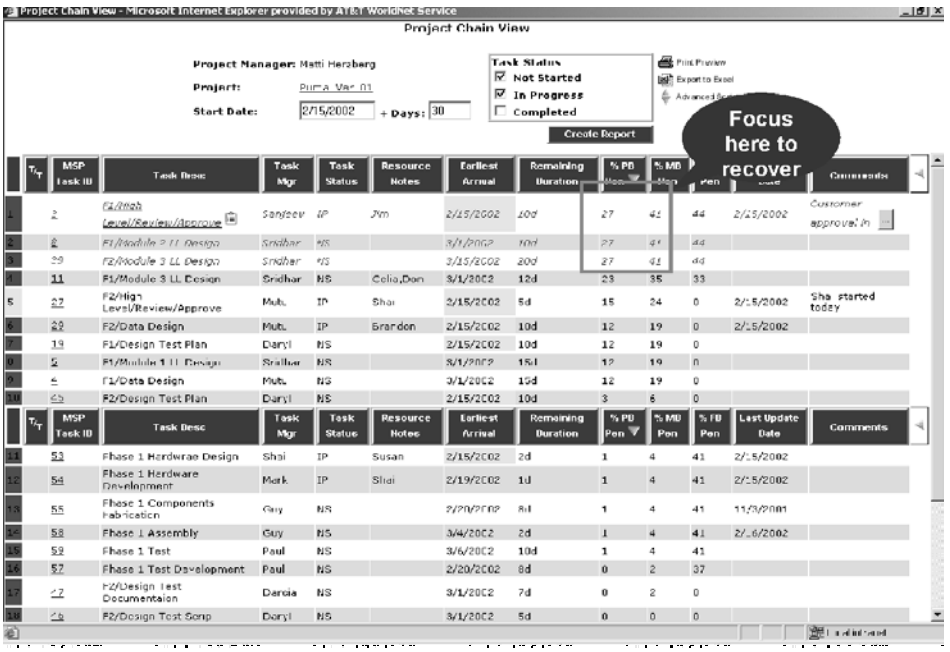


Figure 8.4 The Concerto chain view aids project teams in planning buffer recovery.

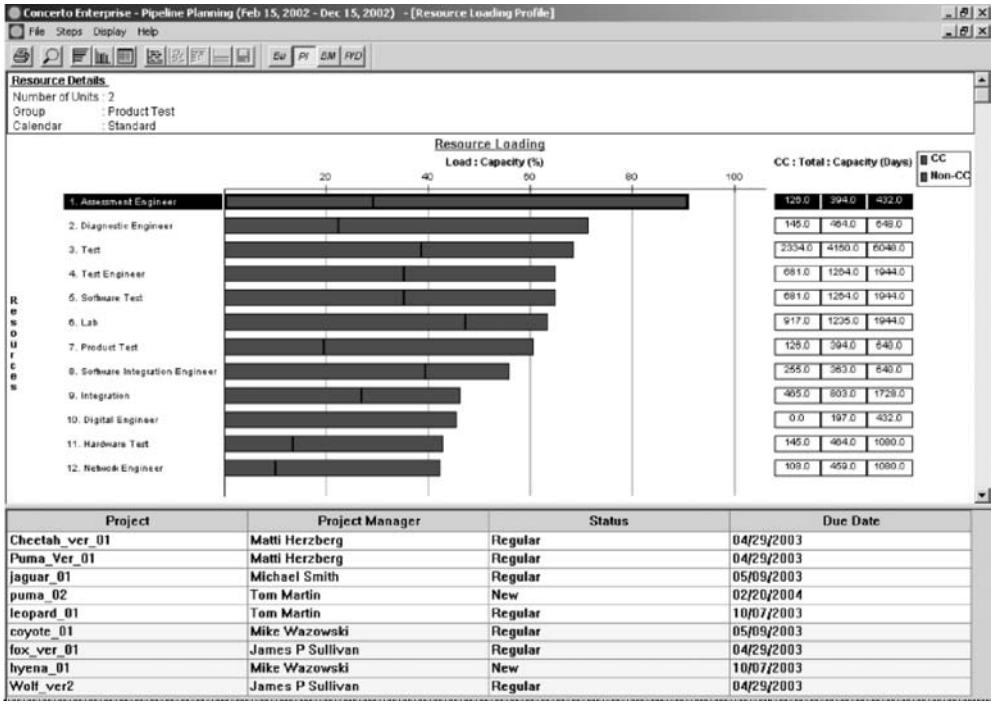


Figure 8.5 The Concerto software's resource portal informs resource managers of the future load expected for their resources.

8.2 Buffer Management

The measurement system for CPPM follows the practice established in Drum–Buffer–Rope (DBR). It uses buffers to measure critical-chain performance. Explicit action levels for decisions minimize making the two mistakes: taking action when you should not and not taking action when you should. Section 6.4.3 recommended explicit action levels for the project manager's primary decision. The project buffer is the most important monitoring tool.

8.2.1 Project Meetings

Project meetings have one purpose: to move the project toward its goal. To achieve that end, meetings must be focused and fast. I recommend that every project have daily, weekly, and monthly meetings focused on moving the project toward its goal.

Project meetings must be highly informative and tightly focused to the needs of the attendees. You should *never* waste valuable meeting time statusing your project schedule. You should never waste all but one meeting participants' time by discussing each task on the schedule. You should not work to resolve problems in general project meetings, unless the problem truly affects everyone in the meeting and requires that everyone be there to solve it. Status your schedule before the meeting, and focus the meeting on necessary communications and assigning actions to resolve blocks to project progress. Meetings should communicate information and identify problems to be solved. The project manager should assign resolution of the problem and move on.

I recommend a fixed agenda and the very rapid publication of minutes following the meetings. The best meeting managers create the minutes as part of the meeting process so that they are available immediately on completion of the meeting. Today, you can easily implement this through online action databases and/or performing the work directly in the schedule program and using a projected display to guide the meeting.

The fixed agenda for the daily meeting usually has only two things on it: a review of the tasks showing yellow and red to ensure that all of the necessary support to recover buffer is working, and a review of the action list to ensure completion of items due that day. The meeting should identify anything that is holding up task completion and assign action to resolve it. Once a week, an additional project meeting might review the project-risk list for additions, changes, deletions, and other key project items, such as project changes or quality issues (e.g., nonconformance reports) and other special items important to your particular project (e.g., long-lead material, contracts, etc.).

8.2.2 The Buffer Report

Clients always want to know how their project is going. Project management sometimes wants to separate the client from the people performing the work for a variety of reasons. The reasons include the clients' disturbing the work flow, workers' mistaking client comments as direction to change the project, and clients' receiving inaccurate information through asking people questions that they don't really know the answer to. (Everybody likes to help.)

Most of us are aware of the organization filter effect. I once had a boss tell me he believed that nothing important got through two layers of management. At the time, I thought him pessimistic. I now realize he was an optimist. Little gets through one layer of management. Information gets distorted as it passes up the chain. Therefore, clients are usually not content with dealing with formal reports or transmissions thorough the formal reporting system.

One of the best ways to keep clients directly informed with accurate information is to invite them to your project meetings. Another way is to give them access to your project-scheduling tool.

Most projects require some type of formal reporting, most often on a monthly basis. These reports are useless for operational control of individual reports. They may be useful for project portfolio decisions (e.g., should we cancel this project?) or for long-term resource-management decisions (e.g., should we hire more resources of a particular type?).

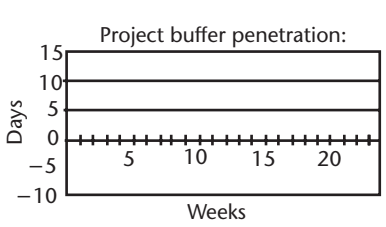
It is much too easy with the computers and sophisticated project-control programs we have now to create very large reports. The cartoon character Dilbert has illustrated the problems with large reports: his boss uses a thick project report as a footrest. Project reporting should help the project, not demand time from otherwise scarce project resources. Therefore, the reports should be very focused on the customer's need for the report. Figure 8.6 illustrates a simple, one-page format for project reporting. The report should contain the minimum information necessary to meet the need and should include a one-page executive summary that tells it all. Figures 8.7 and 8.8 illustrate two ways to display overall progress on a portfolio of projects.

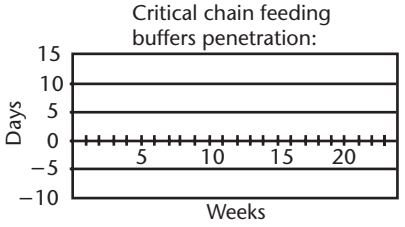
Project: _____ Date: _____

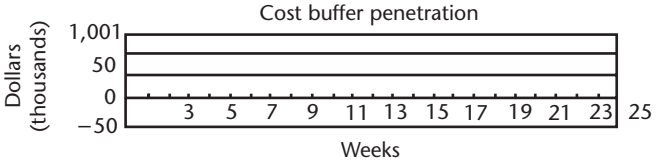
Project Manager: _____

Overal status: ___% of Critical Chain Activities Complete
___% of Critical Chain Activities Planned Complete

Issues or concerns:







Project quality status:

	This month	Previous month	Prior
# of corr. actions needed			
# of corr. actions closed			
Dollars Days Quality			

Change control actions:

	This month	Previous month	Prior
Number submitted			
Number approved			
$ changes			
Schedule impact			

Figure 8.6 Example of project-status report that plots buffer trends.

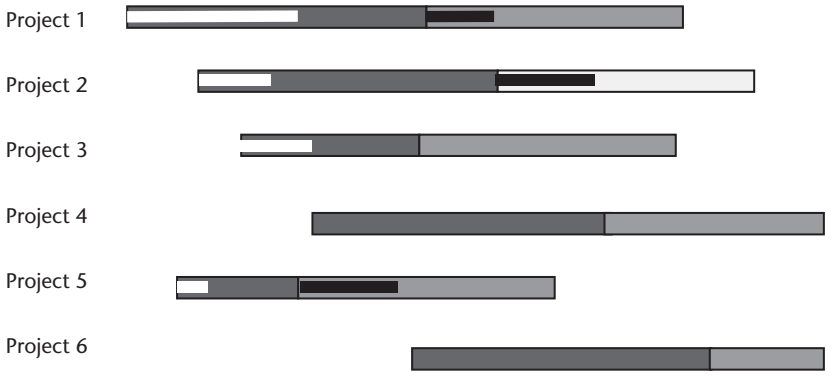


Figure 8.7 Example of schedule progress on a portfolio of projects, illustrating progress and buffer penetration.

Management often overlooks the project team as the recipients of project reports. The project team rarely has the time to read thick reports and often doesn't

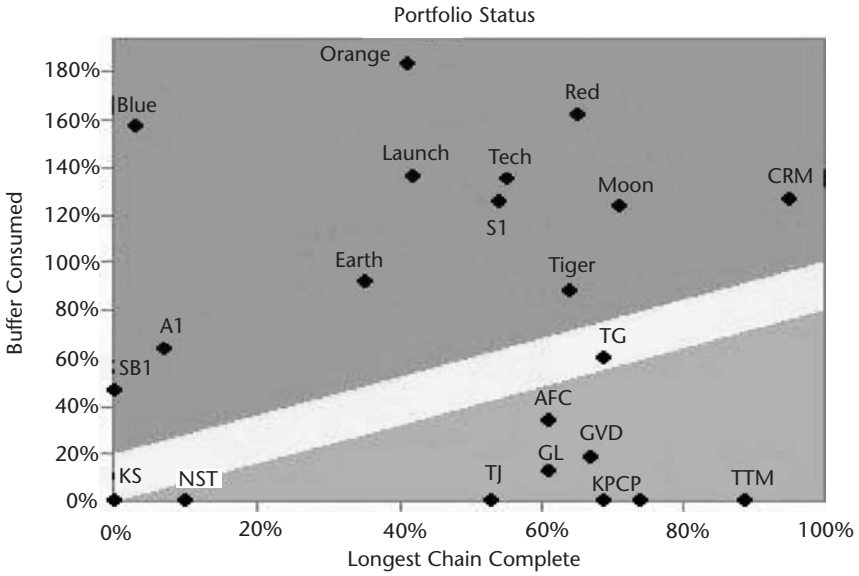


Figure 8.8 Example of schedule progress on a portfolio of projects using the fever chart.

have access to them. There is no excuse for failing to make the information accessible to project participants. The Lean precepts require some type of visual control as feedback to the project team. You should post the statused schedule throughout the project area and have it available on computer networks; you may deploy even simpler measures, such as large signs displaying the days remaining to project completion.

Large projects must report back to the project participants how they are doing (e.g., a weekly all-hands-stand-up meeting). On a large project, you may not want all participants at your monthly project meeting with senior management and the client, but on a small project it may be appropriate to invite everyone.

8.3 Cost Buffer

For many projects, cost is as important as schedule. For some projects, cost may be an absolute constraint. In these cases, it is useful to extend the buffer idea to manage cost to budget. Section 6.5 addresses sizing the cost buffer.

Penetration of the cost buffer provides the global information you need to drive cost decisions. These decisions may include authorizing overtime, bringing additional staff on board, and accepting new work. The measure is cost-buffer penetration in dollars, and the action levels are the same as for the time buffers. You should take no action in the green region of the buffer, plan for actions in the yellow region of the buffer, and take actions when you penetrate the red region of the cost buffer. The cost buffer includes two elements: the net effect of approved project changes and the difference to date between actual and planned cost for the work performed.

8.3.1 Cost Buffer Status

Controlling cost requires measuring it. Most organizations are able to collect actual labor costs on a project with little difficulty. Determining actual material costs can

be another problem entirely. The reason is that material costs may be recognized at different times by different cost-accounting tools. In order to understand cost-buffer penetration, the actual cost must compare directly to the item estimated. In general, you cannot easily accomplish this by simply tracking actual cost versus your time-phased project budget.

Most managers of large projects with significant material costs track material-cost commitments (i.e., the contracted cost for materials ordered). This realization of cost can precede the cost showing up in a project actual-cost report by many months, even years for long-lead-time material.

There is another reason you cannot compare actual project cost to projected project cost versus time to calculate buffer penetration: the actual cost to date includes actual schedule performance, and planned cost to date is based on the scheduled activity performance. A fundamental precept of CCPM is that due to variation, actual schedule performance never matches scheduled performance. To account for this, you can use the earned-value method to determine cost-buffer penetration. As described in Chapter 3, earned value was developed precisely to separate out the two contributors to the difference between cost and estimate on a project: schedule performance and cost performance.

8.3.2 Earned-Value Basics

Earned value defines three terms:

1. Actual cost of the work performed (ACWP): This is simply how much you have spent to date on the project, broken down into the elements of the project. The 2000 PMBOK™ Guide [7] calls this actual cost.
2. Budgeted cost of work scheduled (BCWS): This is the time-phased budget for the project. The 2000 PMBOK™ Guide calls this planned value.
3. Budgeted cost of work performed (BCWP): This is the earned value. You credit activities with a portion (from 0% to 100%) of the budgeted cost for the activity. (Never mind if the activity actually cost more or less than the budgeted amount). The 2000 PMBOK™ Guide calls this earned value.

Note that the above definitions include some different terminology adopted by the 2000 edition of the PMBOK™ Guide. I note these new terms but do not use them from here on as I have not seen them catch on yet.

ACWP is simply the cost to date. BCWS is the budget to date. The only new term here is BCWP, also know as earned value. The difference between cost to date on your project (ACWP) and the budgeted cost to date (BCWS) is the spending variance. Spending variance is made up of two parts: the cost variance (CV) and the schedule variance (SV, discussed below).

8.3.3 Cost-Buffer Penetration

You can use the CV to determine cost-buffer penetration, subject to the understanding clarified below. (A positive CV is good: you are completing more work per dollar spent than you estimated.) A negative CV is positive cost-buffer penetration, and vice versa.

$$CV = BCWP - ACWP$$

This part of earned value can work hand in hand with CCPM, subject to some of the considerations addressed below on determining ACWP.

8.3.4 The Problem

The cost-performance index (CPI) is used in earned value to judge the cost health of the project. A CPI greater than one reflects good cost performance: more work is being accomplished (relative to the estimated cost of that work) than the amount of money being spent. A CPI less than one is bad: less work is being accomplished than money spent.

$$CPI = BCWP/ACWP$$

One can use the initial budget at completion (BAC) and the CPI to calculate the estimate at completion (EAC):

$$EAC = BAC/CPI$$

Note that this approach assumes that the relative underrun or overrun to date will continue to the end of the project. Sometimes this may be a better predictor than just buffer penetration, as using buffer penetration to predict final cost assumes that only the cost overrun or underrun to date will translate to the end of the project.

Most computer scheduling software includes the capability to calculate the earned value, or accumulated BCWP. The ACWP is your actual project cost as of a given date.

The problem is that earned-value thinking is subtly deterministic: it does not address variation or buffers. Figure 8.9 illustrates the overall project budget versus time, considering the project-schedule buffer and project-cost buffer. The project succeeds as long as it completes within the two buffers. The total budget for the project includes the cost buffer. If you sum up the estimated cost for the tasks as BCWP, as suggested by the equation above, the CV does not represent project health relative to the total project budget; that is, a CV of zero on a project that completes on time will underrun the project budget by the full amount of the project buffer. Thus, one way to bring the CCPM and earned value into line is to adjust the BCWP to account for the buffer, as indicated in Figure 8.9. Another way is simply to educate everyone on the project about the revised meaning and to adjust the CPI meaning of "good" to less than one to account for the cost buffer.

8.3.5 Labor Costs

Some organizations substitute other measures than dollars for earned value (e.g., person-hours or person-days). The U.S. Navy does this and substitutes the terminology quantity (Q) for cost in the above acronyms. For reasons that will be clarified below, relating earned value only to labor provides a more useful operational measure.

Most projects have little trouble coming up with comparable BCWP and ACWP for direct-labor costs on a daily basis, sometimes with a little lag. Strict TOC

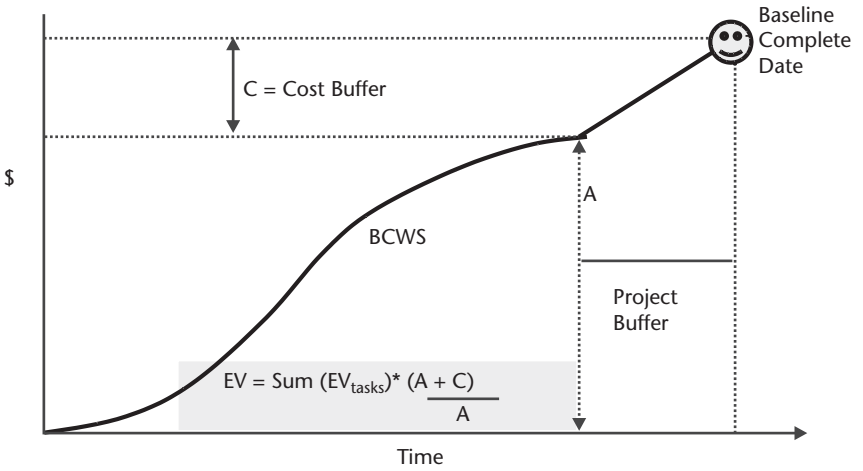


Figure 8.9 Project budget, showing schedule and cost buffer.

thinking identifies a problems with labor costs, first because labor is usually a relatively fixed operating expense and, second, because it includes overhead, and General and Administrative (G&A) mixes in meaningless allocations. But, as a progress tool, we can consider the unit of measure just as a comparative and not worry about these details. A more serious issue lies ahead. Many projects and companies do have trouble with achieving comparable values for material costs.

8.3.6 Material Costs

Few companies are yet able to compile effective actual-cost reports more frequently than monthly and sooner than a week or two after the end of the month. Time lags may be greater for subcontracted work. Unless a project is very long, a significant portion of the project time or budget may be expended before the project manager sees it in cost reports. Multiyear government projects have to work to annual budgets as well as overall project budgets; thus, a six-week delay can represent 10% of the annual budget.

Material costs may include contract labor. The reason project-control systems have difficulty is that the financial systems often lag behind actual material expenditures. You must assure that you account for this in determining cost-buffer penetration. The problems are accruals and commitments.

The accrual problem occurs because you often do not get billed for long-lead-time materials as they are built by the supplier; you usually get the bill upon delivery and take a month or more to pay for it. Your schedule system usually spreads the cost for the material over the time between placing the order and its delivery, which is sometimes many months. Your financial system does not account for the cost until it is paid in one lump sum sometime after the actual delivery. To account for this, some companies estimate "accruals" and include them in the project ACWP. Accruals are estimates of what you owe on the material. Unfortunately, accrual systems are notoriously inaccurate and often have a delay of their own.

Material commitments are the total value of signed contracts not recognized as costs in your accounting system. You may have budgeted $10,000 for some piece of equipment and then had to sign a contract for $15,000 because that was the best

price you could get at the time you placed the order. Your CV should include this difference as soon as you sign the contact because your project will see the cost. In most financial systems, you will not see this difference until the costs are accrued over time and/or until the payment is actually made. Some project-management systems prevent you from changing the budget to account for this difference. You may have to account for this difference separately between committed material cost and actual material cost and add it to your cost-buffer penetration.

8.3.7 Peaceful Coexistence of Buffer Reporting and Earned Value

You can manage the cost buffer the same way you manage the schedule buffer. Figure 8.10 illustrates the cost buffer fever chart, plotting the percent of cost buffer consumption vs. the % of task budget completed. BAC = Budget at completion for the project, including the cost buffer. The % of cost buffer consumed it the CV divided by the cost buffer, expressed as a percent. The % of the task budget complete is the earned value (BCWP) for completed work, as a percentage of the total BCWP for all project tasks. You can use any of the conventional earned value methods to accumulate BCWP for partially completed tasks, e.g.:

- No value until task is complete.
- 50% of value at task start, 50% at task completion.
- An estimated proportional amount.

Since a relatively small number of tasks should be working at any time, it usually makes little difference which method you choose.

You can also create a multiple project cost buffer fever chart, similar to the Figure 8.8 schedule buffer fever chart.

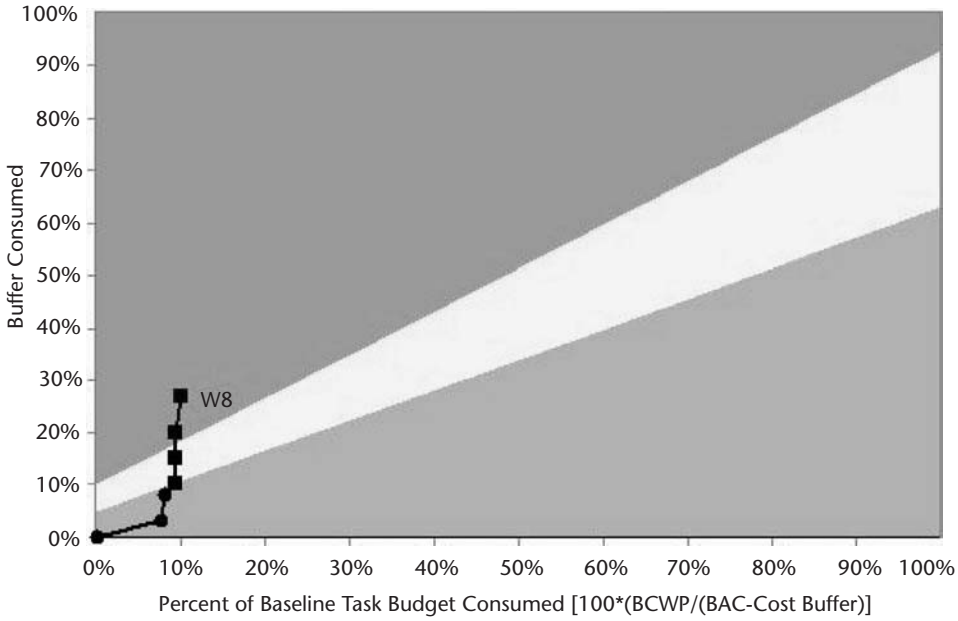


Figure 8.10 Cost Buffer fever chart provides cost tracking and decision information.

8.3.8 The So-called Schedule Variance

Earned value systems also attempt to use earned value as schedule-status data. This requires processing the activity status data with the budget file. Normally, this cannot happen more frequently than the accounting system runs, so sometimes it is monthly, sometimes weekly. This delay is a problem for many projects, as schedule status presented this way is history. Attempting to use such data for operational project decisions is equivalent to driving your car by looking through the rearview mirror.

The earned-value literature also discusses the SV and SPI as companions to the CV and CPI. In my opinion, these terms are misnomers and have nothing to do with schedule. They are not useful to controlling the schedule and can lead to poor operational decisions. They are defined as follows:

$$SV = BCWP - BCWS$$

$$SPI = BCWP/BCWS$$

The so-called SV and SPI actually measure cost, or in some cases, quantity of work input. They do not recognize the critical chain (or critical path) and, thus, do not relate to answering management's question, when are you going to be done? You cannot use them to predict when the project will finish. By weighting cost, they sometimes encourage resources to work on the more costly items, not necessarily the most urgent items to complete the project. A proud construction manager from the world's largest construction firm relayed this behavior to me as his strategy to improve cash flow and profitability on projects. I do not recommend it.

The earned-value "schedule" formulations also do not exhibit the right behavior to be useful for determining when to act to recover schedule. A project that finishes exactly on schedule has an SV of zero and an SPI of one. A project that takes twice as long, when complete, has an SV of zero and an SPI of one. These metrics seem to have no relationship to the actual schedule. Use the CCPM buffer-penetration metric for schedule measurement and control.

8.4 Quality Measurement

Lewis Ireland [8] describes the fundamentals of project-quality management. CCPM does not directly affect the requirements or processes necessary for project-quality control. TOC places a premium on process and product quality because of the importance to the company goal. TOC is a process of ongoing improvement. Quality systems such as those prescribed in ISO 9000 are completely compatible with CCPM.

In *The Haystack Syndrome* [1], Goldratt described an effective measure of quality as "dollar days." Dollar days are the accumulation of the dollars of impact for each day an item does not meet the requirements. Although I have yet to find any company using this measure, I find the logic compelling and describe it for your consideration.

Dollar days put the correct focus on the quality of the product. The technical measures of product quality do not relate directly to either the project or the company goal and necessary conditions. You measure technical performance by conformance to customer requirements, as defined in the customer requirement lists and/or other specifications, standards, or sources of product requirements. These diverse measurement units do not give us an understanding of the importance of quality.

The quality-assurance function usually has the lead to measure and report on quality conformance to plan, but a preventive approach to quality requires all project participants to plan how to not make a defective product. You need a measure that correctly portrays the cost of poor quality and incentivizes the prevention of quality defects. Dollar days encourage quality performance on a project. If one activity must reject an input that does not meet quality standards, the product goes to the quality department, which accumulates the dollar days until passing them on to the activity that caused the defect. (The activity that rejected the product does not get credited with dollar days.)

Dollar days can help us provide the necessary incentive to produce quality deliverables from each activity. The dollar days are passed on to the successor activity as soon as they complete and pass on their work result. The dollar days continue to grow until the excess activity time recovers. Downstream activities will realize it is not fair to penalize them for the duration overruns of predecessor activities but will nevertheless, in most cases, be a little more motivated to get their activity done and pass on the hot potato.

What are the dollars to assign to the dollar-day computation? Several choices come to mind. The daily burn rate of the activity is the low end. The overall project daily benefit is the high end. For activities on the critical chain and for activities on the feeding chains, once they have consumed the project buffer, the total project daily benefit seems appropriate. This will cause immediate focus in the right place. For activities on feeding chains that have not used up the project buffer, the value of the chain seems appropriate.

8.5 Responding to the Buffer Signals

8.5.1 Schedule Buffer Exceeds Yellow Threshold

This is a signal that the schedule buffer may be violated, affecting the overall project schedule. At this level, you must plan ways to recover the schedule on current or downstream activities on the chain. There are three general ways to reduce activity time: increase resources, reduce scope, or improve the process for the activity. Table 8.1 lists ideas to help reduce schedule-buffer penetration.

8.5.2 Cost Buffer Exceeds Yellow Threshold

This is a signal that the overall project may overrun the budget. You must plan ways to reduce cost. Depending on the trend and the indications and projections from the cost-control chart, you may initiate action before exceeding the second third of the cost buffer. Table 8.2 lists ideas to help reduce cost-buffer penetration.

Table 8.1 Ideas to Help Reduce Schedule Buffer Penetration

Methods to Increase Resources	*Methods to Reduce Scope*	*Methods to Improve the Process*
Add additional staff	Subcontract part of the scope	Change the activity logic (e.g., go from finish to start to finish to finish)
		Examine the activity logic for ways to reduce batch sizes
Break up the activity to use a more diverse kind of staff	Revise requirements	Provide improved tools
Pay overtime (for labor)	Defer requirements to later in the project	Obtain expert assistance
Use subcontract labor		Use process improvement tools, especially cycle-time analysis
Add incentives (for subcontracts)		Perform some of the work early on downstream tasks to reduce their duration

Table 8.2 Ideas to Help Reduce Cost Buffer Penetration

Methods to Reduce Cost	*Methods to Reduce Scope*	*Methods to Improve the Process*
Use lower-cost staff	Subcontract part of the scope	Change activity logic (e.g., go from finish to start to finish to finish)
Use more-productive staff	Revise requirements	Provide improved tools
Use competitive bidding for subcontracts	Defer requirements to later in the project	Obtain expert assistance
Perform make-buy analysis on planned subcontracts and on activities that might be subcontracted at reduced cost	Look for activities that can be deleted	Use process-improvement tools, especially cycle-time analysis
Use lower-cost supplemental staff to off-load high-cost staff of routine duties	Look for costs that may not be necessary to meet the customer's requirements	

8.5.3 Dollar Days' Quality Increasing

This means that our quality process is not effective. You need to perform problem solving to discover and correct the Core Problem. You must have the quality process well defined and in control for problem solving to be effective.

8.5.4 Schedule Buffer Exceeds Red Threshold

The first thing that should happen is that resources place top priority on tasks in a chain that is causing red zone buffer penetration. This includes all modes of exploiting and subordinating, starting with over time on such tasks.

This is the signal to implement the action you had planned. Depending on the changes necessary to implement recovery, you may need to adjust the project plan, using your formal project-change-management procedure. If you change the project logic, such as, for example, going to finish-to-finish logic instead of start-to-finish logic, you should change the plan accordingly. Changes such as authorizing overtime or using contract labor should not require a change to your plan.

8.5.5 Cost Buffer Exceeds Red Threshold

This is the signal to implement the action you had planned. You need not change the plan for cases where actual cost simply exceeded the estimate and you intend to absorb them into the cost buffer. As above, if your plan requires changes in the project logic or scope, you should implement a formal project change along with implementing the action.

8.5.6 Schedule or Cost Buffer Exceeds 100%

If you have created a plan to recover buffer by the end of the project, you need not take additional action when this happens. If you do not have a plan to recover the excess buffer by the end of the project, you need to put in a project change. You probably also need to recalculate the project going forward and to reinstitute a realistic amount of buffer. Once the buffer penetration has exceeded 100% and you have no opportunity to recover, the buffer loses usefulness as a control tool.

8.6 Milestones

Somehow some people have seized on a belief that CCPM is opposed to milestones. I am not sure where this erroneous belief originated, but I suspect it might come from misinterpreting comments made by Goldratt opposing unbuffered fixed-date milestones. Section 5.6 presents my view that milestones are a powerful project-management tool. On longer projects, they help enormously to focus and reward the work of the project team. I always try to have a major milestone accomplishment at least every quarter.

CCPM supports two kinds of milestones: floating milestones and fixed-date milestones. Fixed-date milestones must have a buffer preceding the date. Size the buffer as a project buffer for the chain leading to the milestone. You do not need to use that buffer for measurement: it is there to give you a high-probability milestone date.

Floating milestones go into your network and represent key technical accomplishments. They should not have a date associated with them, either planned or actual. They are a great aid to network development for marking completion of major project accomplishments (e.g., completion of design) and are therefore also great accomplishments to celebrate.

8.7 Change-Control Actions

Section 5.11 describes the need and process for formal project-change control. You might as well embrace project change because the reality is that project plans change all the time. The project manager should approve any changes to the plan. You have to decide on the criteria that constitute a formal change to the plan. You should have a form or computer process to track changes and a configuration-control mechanism to assure that everyone works to the latest version of the plan. The following are some thoughts for your consideration:

- A change in the plan logic (e.g., adding a task, deleting a task, making a change to the predecessor or successor on a task) should comprise a change.
- A significant change in the scope of a task (you have to define *significant*) should be considered a change.
- A significant change in the task resource requirement or in the identification of the resource should be considered a change. It may be necessary to recheck the critical chain.
- Overrun or underrun of task-duration estimates is not a change.
- Overrun or underrun of estimated task cost is not a change.
- Project-, feeding-, and cost-buffer action triggers may cause you to change the plan to recover buffer.
- Your change-control process should operate quickly. You may have a change-control board, including your customer when appropriate, to expedite change approval.

Keep in mind that you should focus on managing the project to the plan, not on managing the plan. Do not, for example, make changes to your buffers based on actual performance to date.

8.8 Frequently Asked Measurement-and-Control Questions

1. We are halfway through the project, and have not penetrated the project buffer. Can we cut the project buffer in half?

   Cutting the project buffer does not reduce the project's actual performance time. It reduces the chance that the project will deliver substantially early. Your project-buffer status gives you dynamic predictions of the project's completion time. There is no reason internal to the project to reduce the project buffer. It wastes resources for no real benefit.

   If external needs require you to reduce the project buffer, you can replan the project at any time. Remember that the project buffer protects the whole project. All noncritical chains merge with the critical chains before the project is complete. You should check all feeding buffers to ensure that the unused feeding buffer length is at least 50% of each feeding-chain uncompleted path length before you reduce the project buffer. If the feeding buffers are all intact by this amount, there is no problem with reducing the project buffer to 50% of the remaining length of the critical chain. In essence, you are starting a new project at the time of the update.
2. Why does our prioritized task list change from day to day?

   Your prioritized task list can change every time you update schedule status. Ensure that people understand they should continue working on a started task until they complete it, then look to the task list to pick the next task to work on. This will assure the most rapid project completion.
3. The prioritized task list says I should work on a lower-priority project's task before I work on the task for a higher-priority project. How can that be?

The task list is probably right. This happens when the chain containing your task on the lower-priority project has greater project-buffer penetration than the chain your task is on in the higher-priority project. Recall that project priority was made implicit in the project schedule by project pipelining. Your work on the higher-priority project may not impact project completion when the task chain your task is on is not the path with most buffer penetration for that project. Your task completion might directly affect the completion date of the lower-priority project. Therefore, to make a local decision that supports the flow of all projects for the company, you should follow the task priority.

4. I haven't even started work on my task yet, and it's in the red zone of the buffer. Why should I be blamed for someone else not doing his or her job?

   Project management must make it clear throughout the organization that this is not a matter of blame. All need to focus on fixing the cause of the problem and not waste a nanosecond on fixing blame. People should view a red task as a ticket to focus all available effort on completing the task as soon as possible. It is a signal for priority handling. Everyone must understand that a task can have its buffer in the red zone because of delaying tasks upstream of it. The task may be (hopefully is) recovering buffer.

5. Why does the prioritized task list come up with tasks as high priority that we know are less important than other tasks?

   This result may indicate a problem with the modeling of the project. It may also be more or less random, a result of the necessity of adjusting the resource demand to the resource supply. You should use your judgment when responding to the task priorities offered by the model. Be sure to check and understand why the task is being offered as a higher priority before you move down the list. You may be surprised! Also, sometimes certain tasks are easier to work on than others, and resources will choose to work on lower-priority tasks, hoping someone else will get the less-desirable task. You need to make sure this does not happen.

8.9 Summary

CCPM uses buffer consumption and the rate of buffer consumption (percentage of buffer consumed versus percentage of critical chain complete) as the primary, real-time, predictive measurement tools. You should consider both clients and project-team members as customers of your project-reporting-and-control system. Since buffer reporting must be timely to be effective, you should status and report buffers daily and ensure that the information is immediately available to all users. The most important items in this chapter include the following:

1. Daily buffer monitoring and reporting provides a proactive, real-time decision tool for project control. Task managers use the information to decide which task to work on next. Project mangers use the information to decide when to take action on a project. Resource managers use the information for longer-term resource decisions.

2. Focused project meetings, using previously statused schedules, are a powerful tool to move your project to successful completion.
3. Buffer-recovery planning and execution are essential parts of the CCPM control process.
4. If cost is important to your project, you can use the buffer and earned value (BCWP) to derive cost-buffer penetration, but do not attempt to use earned value for schedule control.
5. CCPM puts a premium on project-quality management.
6. Conventional project-change-control methods are necessary to handle scope changes and the impacts of special-cause variation.

CCPM users are finding implementation of buffer management to be relatively simple and a very effective overall approach to project management and control.

References

[1] Juran, Joseph J., *Juran on Planning for Quality*, New York: The Free Press, 1988.

[2] Goldratt, Eliyahu M., *The Haystack Syndrome*, Croton-on Hudson, NY: North River Press, 1990.

[3] Herroelen, W., R. Leus, and E. Demeulemeester, "Critical Chain Project Scheduling: Do Not Oversimplify." *Project Management Journal,* Vol. 33, No. 4, December 2002, pp. 48–60.

[4] CCPM+ Software, available at http://www.advanced-projects.com/CCPM+.htm (accessed June 2004).

[5] Concerto Software, available at http://www.Realization.com (accessed June 2004).

[6] Shewhart, Walter A., *Statistical Method from the Viewpoint of Quality Control,* New York: Dover Publications, 1986 (originally published in 1939).

[7] PMI, *A Guide to the Project Management Body of Knowledge*, Upper Darby, PA: PMI, 2000.

[8] Ireland, Lewis R., *Quality Management for Projects and Programs*, Upper Darby, PA: PMI, 1991.

CHAPTER 9

Implementing the Change to CCPM

Many companies that have never introduced change into an organization successfully implement CCPM in a short time. These successful companies require less than three months to get all projects planned and synchronized and to begin seeing the benefits of improved project performance and reduced stress on project teams. Success stories include all types of projects and organizations of a wide range of sizes.

People have reported significant success implementing CCPM for single projects after reading an earlier version of this book or having attended a two-day introductory training class. Unfortunately, some organizations claim to have attempted critical chain and, for one stated reason or another, given it up. That has happened on both single projects and in multiproject organizations. Other organizations (generally larger ones) can take a year or longer to get all of the projects planned as CCPM projects. The following presents a process proven to work in both single- and multiproject organizations, and then presents the theory underlying the approach.

9.1 Implementation Model

Figure 9.1 illustrates the basic project model to implement CCPM. Implementation is a project. Implementation plans vary in content and scope due to the specific organization. The rest of this section presents an example implementation plan. All implementations should include all of the steps outlined here, but you should adjust the plan's depth and breadth to the unique needs of your organization. Following the example plan, Section 9.2 describes some of the theory underlying it.

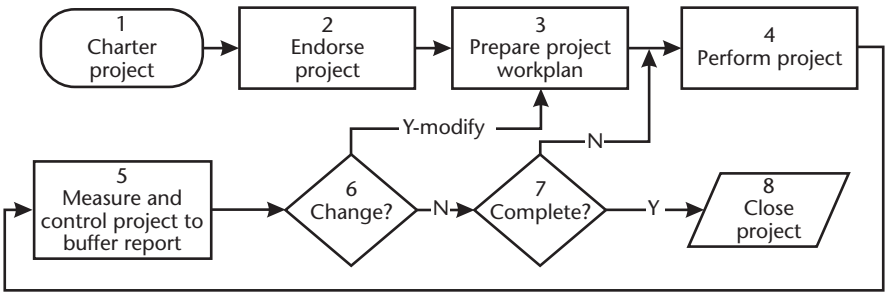


Figure 9.1 Implementation process flowchart.

9.1.1 Endorse the Implementation Project

J. Kotter's [1] first two points of successful change programs are to establish a sense of urgency and to create a powerful guiding coalition. You can link these two elements as you gain endorsement for the implementation project. You have to find out who is willing to lead a change in the way the organization performs and come up with a reason for these people to want to put forth the necessary leadership. Kotter notes that well over 50% of changes fail in the first step: not establishing enough sense of urgency. Usually, the higher up you go in an organization, the more likely you are to find a sense of urgency to deliver the quantitative results that CCPM offers. Since multiproject CCPM requires the collaboration of project and resource managers across all functions, establishing a powerful guiding coalition is probably more important for multiproject CCPM than for many organization change initiatives.

Endorsement means getting the stakeholders to agree in the beginning that they are willing to assist as necessary to effect the change to CCPM. While in some instances it may be enough to have people say, "I don't have a problem with that," in most cases you need more than permission. You need a willingness to change on their part. Stakeholders include the project teams, project managers, resource managers, senior management, clients, and suppliers. There may be more stakeholders important to your implementation, perhaps even stockholders. Sometimes, you may wish to obtain this endorsement before you have the project charter. In other cases, you may wish to use a draft of the project charter as your vehicle for endorsement.

Kotter addresses the often-stated premise that major change is impossible unless actively supported by the head of the organization. That is certainly my experience with implementing CCPM. But, I have also noted that in today's organizations, active support by the formal organization leader is rarely enough. Kotter notes,

> In successful transformations, the chairman or president of the division, general manager, plus another 5 or 15 or 50 people, come together and develop a shared commitment to excellent performance through renewal. In my experience, this group never includes all of the company's most senior executives because some people just won't buy in, at least not at first. But in most successful cases, the coalition is always pretty powerful—in terms of titles, information and expertise, reputations and relationships. (p. 6)

My experience with implementing CCPM fully supports Kotter's assertions.

9.1.2 Charter the Implementation Project

The following project charter provides a sample critical-chain implementation project charter. Try to keep yours to one page and focus on the needs of the project stakeholders.

9.1.3 Begin with the End in Mind (Vision)

If you don't know where you are going, you won't know when you get there. You may represent the end vision a variety of ways. Kotter's third point is to create a vision of the state that will exist after the change is made, when the organization is functioning using CCPM. A picture usually helps, as many people respond better to

Project Charter

Project: Implement Critical Chain Project Management

Revision: 0 Date: 1/1/2005

*Approved by:*_____

Project purpose:
The Critical Chain Project Management (CCPM) implementation project will install CCPM for management of all projects performed by the southwestern division of ACME Products Supply Corporation.

Customer and stakeholders:
The primary individual customer for this project is Wiley E. Coyote, director of ACME Products, southwestern division. The customer group is all employees, including managers, of the division. Client-customer involvement, such as R. Runner, can be included in this project if client involvement is necessary to implementation.

Project team:
Cynthia Standish is the project director. She will select three to five team members, as necessary, to assist in planning, scheduling, and other implementation project activities. All managers within the division are to support the implementation project as required.

Scope:
This project includes all of the planning, procedure development, training, and software tools necessary and sufficient to install CCPM into the division. It does not include technical work on the projects or work with the project customers.

Schedule:
The use of CCPM is expected to be substantially complete within 90 days of the approval of this charter. Quarterly progress reviews are to be held for the following three quarters (i.e., the final one is to be held on February 6, 2006).

Cost:
The overall cost of this project, including expenditures for training (not including employee time), consulting support, procedure development, and the software tool shall not exceed $250,000, without additional management authorization. Cost associated with the replanning of projects using CCPM and buffer management are not included in this cost as they are part of the respective projects.

Special considerations:
Procedures and software tools should comply with company format and computing capability.

*Acceptance:*_____, *Project Manager*

visual stimuli than to other inputs. Consider putting together your own picture of your organization operating under the critical-chain paradigm. For engineers, this picture may look a lot like a diagram. To project managers, it may be a picture of a simplified Gantt chart, showing the features of critical-chain plans. To "people people," it will include people. I prefer to describe it in terms of the behavior necessary to operate critical-chain project successfully, because it is all about changing behavior. Many organizations make the mistake of thinking it is about buying and implementing software. It isn't.

Kotter's fourth and fifth points are to communicate the vision and empower others to act on it. A vision can only be effective if it is communicated. Everyone affected by the vision should be able to state immediately the expected outcomes of implementing CCPM. Communicating the vision entails empowering others to work to act on the vision, including overcoming the obstacles to change. Table 9.1(a–e) lists the various behavior changes to be expected from different groups.

Table 9.1(a) Senior Management's Behavior Changes

Change	*Present Behavior*	*Future Behavior*
Only commit to feasible delivery dates	Sometimes commit to arbitrary delivery dates determined without consideration of system capability to deliver	Only commit to delivery dates with a critical-chain plan and, on multiple projects, after sequencing through the drum schedule
Eliminate interruptions	Insert special requests to the system with no assessment of system capability to respond Sometimes place demands for routine administrative work above project work (e.g., salary reviews)	Prioritize all requests using buffer report
Set project priority*	Lack clear project priority or have changing project priorities	Set project priorities, including the priority of new projects relative to ongoing projects
Select drum resource*	Fail to consider system constraint	Select the drum resource to be used for sequencing the start of projects and creating the drum schedule
Select drum manager and approve project sequencing*(pipelining)	Start each project independently as funding is available, or start all projects at the beginning of the year.	Drum manager creates drum schedule Senior management approves Project managers schedule projects to the drum
Report project status	Look over shoulders	Task managers estimate ROC Prepare buffer report

* Only for multiple projects.

Table 9.1(b) Resource Managers' Behavior Changes

Change	*Present Behavior*	*Future Behavior*
Assign resource priority	Assign resources on a "first-come, first served" priority basis or attempt to meet all needs by multitasking	Assign resources using the buffer report
Do resource planning	Plan resources by name and task	Plan resources by type and assign them to tasks as they come up, using the buffer report to determine priority
Complete early	Turn in tasks on due date	Turn in tasks as soon as they are complete
Eliminate multitasking	Assure resource efficiency by assigning them to multiple tasks at the same time	Assure resource effectiveness by eliminating bad multitasking
Generate resource buffers	Schedule resources far ahead that are then not available when needed	Use task priority list to dynamically assign resources to tasks based on buffer penetration.

9.1.4 Create the Implementation Project Work Plan

The project work plan is the next step following the project charter and includes

1. Detailed specification of the project scope;
2. WBS to organize the project scope;
3. Assignment of responsibility to the WBS;

Table 9.1(c) Project Managers' Behavior Changes

Change	*Present Behavior*	*Future Behavior*
Use 50% task-duration estimates	Project managers send a message that they expect due dates to be met	Project managers first derive low-risk task duration and then get average duration, using task uncertainty to size buffers
Develop relay-racer task performance	Provide start and finish dates for each task and monitor progress to finish dates	Provide start dates only for chains of tasks and completion date only for the project buffer
Control feedback on task-duration overruns	Management provides negative feedback when tasks overrun due dates	Management provides positive feedback and help if resources perform to relay-racer paradigm
Determine project status	Varies: often use earned value as the schedule measure	Use buffer report (including a cost buffer)
Allow project changes	Varies: often submit changes to minimize minor variances	Allow when triggered by buffer report
Respond to management demands for shorter schedule	Arbitrary task-duration cuts	Add resources or make process changes to get a feasible schedule immune from common-caused variation
Start early	Start tasks as early as possible	Start task chains as late as possible, buffered by feeding buffers
Sequence projects*	Start project as soon as funding is available	Schedule project start using drum schedule
Assign resources dynamically according to critical-chain priority and buffer reports	Get resources as soon as project funding is available and hold resources until they can't possibly be used any more on the project	Get resources only when needed and release them as soon as task is complete

Table 9.1(d) Subcontractors' Behavior Changes

Change	*Present Behavior*	*Future Behavior*
Deliver to lead-times	Deliver to due dates	Deliver to lead-times
Shorten lead-times	Deliver to due dates	Shorten lead-times

Table 9.1(e) Customers' Behavior Changes

Change	*Present Behavior*	*Future Behavior*
Minimize project-scope changes	Spend little time initially establishing requirements and then introduce late changes	Establish requirements as part of the project work plan and change as little as possible with formal change control
Support using project buffer	Interpret contingency as "fat"	Understand the need for buffers to reduce project lead-time and ensure project success
Eliminate arbitrary milestone dates	Demand arbitrary milestone dates	Use plan to set milestones Procede all dates with a buffer

4. Resource-loaded (critical-chain) project schedule;
5. Project budget;
6. Definition of the project team;
7. Procedures for operation of the project team;
8. Plans for project close out.

Figure 9.2 illustrates the WBS for the project to implement CCPM. The WBS reflects the changes necessary for CCPM and includes the overall approach

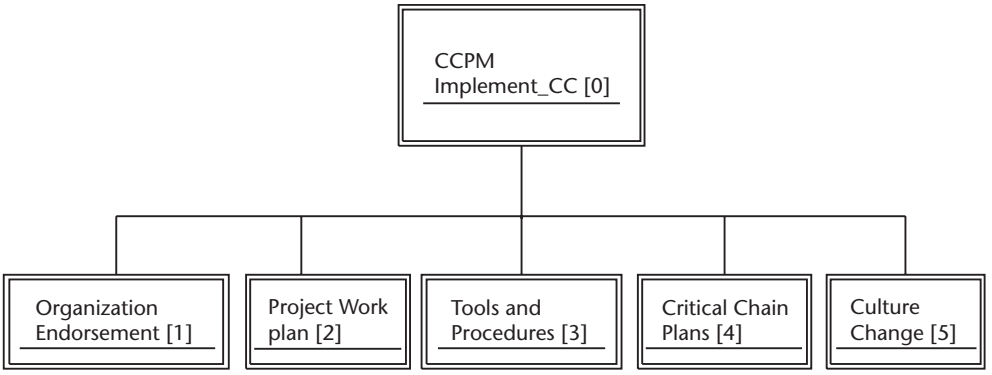


Figure 9.2 The WBS to implement CCPM identifies the work package deliverables.

recommended by Kotter [1] (See Section 9.2). Kotter's sixth point for effective change programs is to create short-term wins for those who participate early in the change. Be sure to make this part of your change plan.

The necessary actions to create a CCPM organization consider both resource behavior and the technical requirements of critical chain, including the following:

1. Project plans follow the CCPM paradigm (i.e., 50% task times, critical chain, and properly sized buffers).
2. The drum manager creates the drum schedule to accommodate management's project priority.
3. Project managers schedule projects to the drum schedule.
4. Resources work to the relay-racer paradigm.
5. Resources provide accurate input on the RDU of their tasks.
6. Resources and resource managers use the prioritized task lists to avoid bad multitasking and to determine which task to work on next.
7. Project managers plan buffer recovery when the project buffer enters the yellow zone and implement the plans when buffer penetration enters the red zone.

The work-plan tasks developed following the WBS must create these results.

You should consider the Seven Ss and the actual behavior in your (see Section 9.2.1) organization in order to complete your plan. Try to separate fact from fiction. For example, many people initially believe that all tasks in their organization are underestimated. This is often the case in an organization with extensive multitasking and interruptions. Sometimes, they have data on actual, reported task completion for previous projects. It usually only requires a quick check to find that they are similar to most organizations, showing extensive date-driven behavior.

Note that I have focused the WBS on the required behavior change and not on changes in the underlying beliefs or culture. This accomplishes the most direct change. It may not lead to lasting change. While some of the feedback from CCPM is self-reinforcing, it is legitimate, in many organizations, to fear that management's exploitation of this method will extend to exploitation in a negative sense, such as overloading the drum resource. If your organization is misaligned with the principles underlying the TOC, you should take this opportunity to begin the cultural and

belief changes needed for ongoing improvement. Otherwise, improvement will stop as soon as the implementation project ends.

The responsibility matrix identifies each of the work packages shown in the WBS and the person responsible and accountable for the work package. You may assign responsibility at multiple levels in the WBS, but you *must* assign it to the lowest, or work-package, level. Note that the person responsible for the work package is not necessarily the same as the resources required to perform the work contained in the work package. The responsible and accountable work-package manager may be one of the resources that works on the project and may show up in his or her own work package and in other work packages. Work-package managers plan and estimate the work package and then are accountable to manage its performance.

The WBS and project schedule include Kotter's final two points of effective change programs: consolidate improvements while producing still more change and institutionalize the new approaches. The final step requires assuring that the CCPM approach permeates all policies, procedures, and measures of the organization and is formalized into training programs to ensure that new people are properly indoctrinated into the organizational process. In the end, CCPM should not be an additional thing. It should just be "how we do business around here."

Figure 9.3 illustrates a schedule for implementing CCPM. It is in the critical-chain format, as illustrated by the CCPM+ software–modified Microsoft Project Gantt chart. This format illustrates all of the tasks on the project but identifies the critical-chain tasks by color (difficult to see in a gray-scale publication.) It also shows the tasks that may be performed early without causing an adverse resource conflict for the project. Do not start these tasks early in a multiproject environment because you may cause unnecessary schedule conflicts during project performance.

This schedule only shows the end dates of the milestone and project buffer. The plan, starting on January 2, 2005, predicts completion (end of the project buffer) on October 12, 2005. The project and feeding buffers are 50% of the preceding chain. Note that the WBS elements include from four to eight tasks. Eight tasks are by no means the limit for a work package; work packages often have up to 25 tasks.

The pilot project dominates the critical chain. This schedule includes a 60-working-day duration for the pilot project (i.e., about three months). The pilot project usually need not complete to serve its functions, such as identifying the organizational obstacles and serving as a test for the procedures established for full-scale implementation. You should formally declare success on the pilot project when you have accumulated sufficient information that the organization is ready to move on to the next step.

Your plan will differ from this one. If you have a large organization, the number of projects to be planned may be much larger. Training durations will usually be longer due to the need to schedule multiple sessions to allow for people's availability and sometimes to accommodate different locations. Sometimes acquiring the software can take much longer, but you should work to keep that process off the critical chain, including doing the pilot project with your existing critical-path software.

9.1.5 Plan to Prevent or Mitigate Implementation Risks

Project-risk management seeks to control potential causes of special-cause variation of high probability and consequence. The project plan monitors and may include

	WBS	Task Name
1	1	**Organization Endorsement**
2	1.1	Initial Training Session
3	1.2	Establish Guiding Coalition
4	1.3	Define Vision
5	1.4	Create Project Charter
6	2	**Project Work plan**
7	2.1	Create WBS
8	2.2	Assign Responsibility
9	2.3	Develop Key Milestone Chart
10	FB_40H	Feeding buffer
11	2.4	Draft Operation Procedures
12	2.5	Develop Critical Chain Network
13	2.6	Project Kick-off meeting
14	3	**Tools and Procedures**
15	3.1	Purchase Software
16	3.2	Install Software
17	3.3	Set up Resource File
18	3.4	Set up Software Permissions
19	3.5	Software Training
20	3.6	Define CCPM Roles
21	3.7	Draft Procedures
22	3.8	Draft Procedure Training
23	3.9	Train to Procedures
24	4	**Critical Chain Plans**
25	4.1	Select project sequence
26	FB_41H	Feeding buffer
27	FB_42H	Feeding buffer
28	FB_43H	Feeding buffer
29	FB_44H	Feeding buffer
30	4.2	Perform Pilot Project
31	4.3	Plan Project 1
32	4.4	Plan Project 2
33	4.5	Plan Project 3
34	4.6	Est Project Priority
35	FB_45H	Feeding buffer
36	FB_46H	Feeding buffer
37	FB_47H	Feeding buffer
38	FB_48H	Feeding buffer
39	4.7	Pipeline Projects (Drum schedule)
40	FB_49H	Feeding buffer
41	FB_50H	Feeding buffer
42	FB_51H	Feeding buffer
43	4.8	Launch CCPM Projects
44	5	**Culture Change**
45	5.1	Train Project Managers
46	5.2	Train Resource Managers
47	5.3	Prepare Communication Plan
48	5.4	Publicize Short Term Wins
49	5.5	Institutionalize Process
50	PB_H53	Feeding buffer
51	PB_52H	Project buffer
52	5.6	Implementation Complete

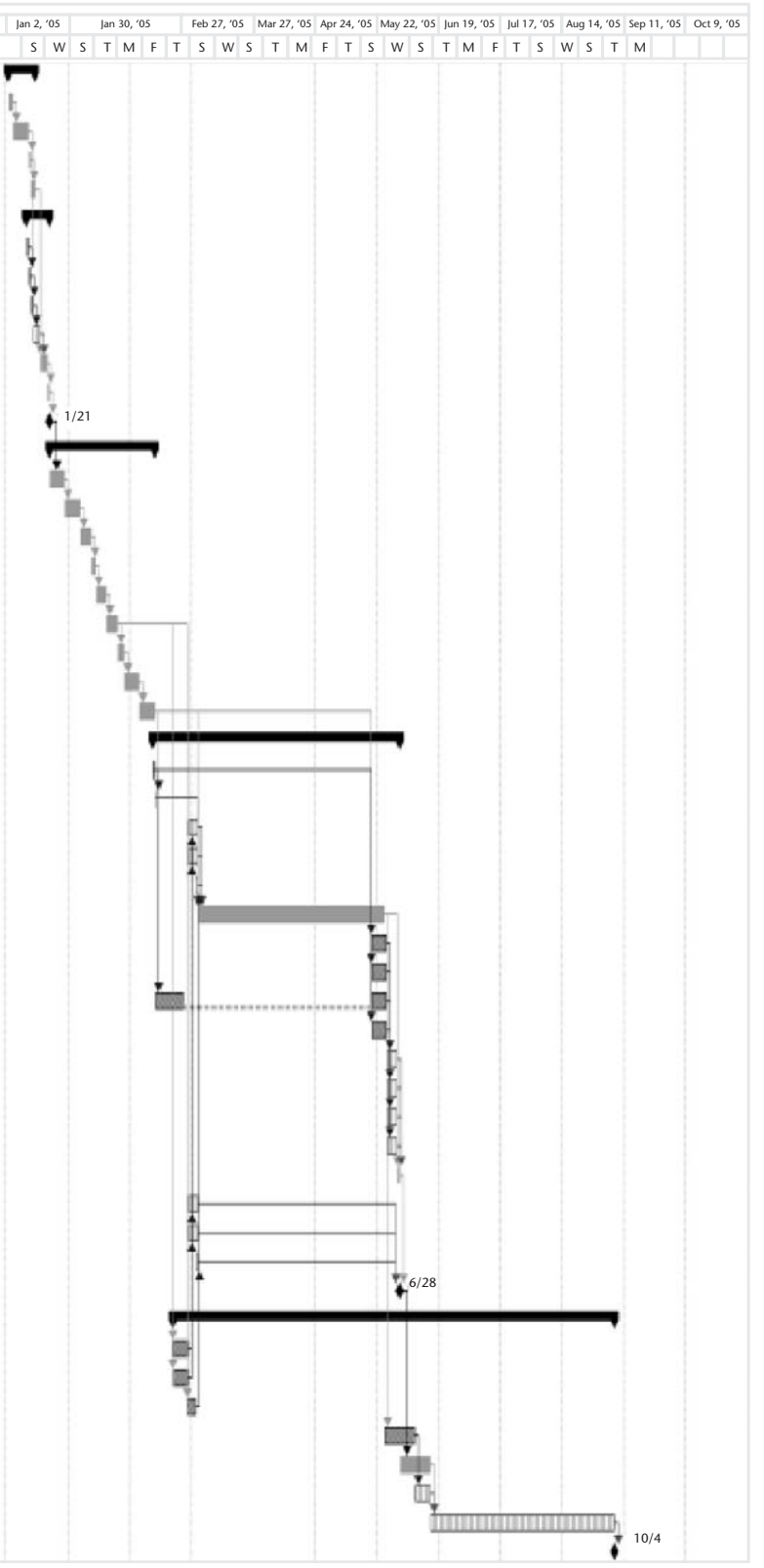


Figure 9.3 Critical-chain plan to implement critical chain, shown in CCPM+ Gantt format.

prevention or mitigation planning for causes of sufficiently high probability and/or consequence. These special causes of variation are very organization specific.

Start by assessing the risk using whatever tools are appropriate to the magnitude of the project and comfortable for your team. Your plan can include a very rudimentary level of risk assessment if your projects are small and pose no health, safety, or environmental risks. The other end of the spectrum may include multimillion-dollar, probabilistic risk assessments performed by teams of Ph.D.-level scientists, engineers, and legal and business experts. Critical-chain implementation is at the low end of this spectrum. It does not impact the success of ongoing projects in a negative way, even if they do not achieve the critical-chain benefits.

Table 9.2 presents an example CCPM risk-management matrix, using the format described in the next chapter for all of your projects. Your risks will differ from this example list; therefore, your plans to prevent or mitigate risks will also be different. Please refer to the table legend at the bottom of the table for understanding the second through fourth columns. Note that, as with all risks, if the probability is greater than 50%, your baseline assumption should be that the risk event will happen. As with all risk-management plans, you should review and revise the plan as the project progresses.

9.1.6 Just Do It! or Fake It Until You Make It

Tables 9.3 through 9.5 lay out the steps necessary to implement CCPM in a large, multiproject environment. Scale these steps to fit your needs.

Your team can accomplish all of the Table 9.3 actions in a one-day meeting. This meeting best follows a two-day workshop where all of the mangers learn the critical-chain theory. It is a mistake to separate the two-day workshop from this

Table 9.2 Example CCPM Implementation Project Risk-Management Plan

#	*Risk*	*P*	*C*	*R*	*Prevention*	*Mitigation*
1	Initial pilot project does not do well.	2	2	4	Apply readiness review to Pilot Project to ensure all necessary conditions are met.	Frequently monitor progress and coach the project manager and team.
2	Senior management does not exhibit CCPM behaviors.	2	3	6	Train senior management. Attain a guiding coalition of senior managers.	Facilitate senior management training and decision sessions.
3	Initial CCPM projects compete for resources with non-CCPM Projects.	3	1	3	Select initial projects with minimal overlap.	Coach resource managers on process to resolve contentions with minimal bad multitasking.
4	External contractors do not work to CCPM paradigm.	2	2	4	Include contractor performance expectations in contracts.	Facilitation sessions with contractors.
5	Resource loaded networks inadequate for projects.	2	3	6	Use experienced consultants to assist initial network building.	Apply change control to revise networks as necessary.

Legend:

P (Probability)	C (Consequence)	$R = P * C$
3 = 20–50%	3 = >Project Buffer	
2 = 5–20%	2 = ≥20% of Project Buffer, ≤Project buffer	
1 = <%%	1 = <20%	

Table 9.3 Implement Phase 1

Step	*Resp.*	*Action*	*Output*
1	Facilitator	Plan a session with the leadership team and inform them of the agenda	Meeting schedule and agenda
2	Facilitator	Brief the team on the multiproject solution	Knowledge
3	Leaders	Leadership team identifies the constraint resource (drum) for the organization	Constraint (drum) resource identified
4	Facilitator	Present (briefly) buffer management	Knowledge
5	Leaders	Assign responsibility for buffer reporting	Responsibility assignment
6	Project Managers	Commit to track and manage to buffers	Commitment
7	Leaders	Select initial project for CCPM	Project list
8	Leaders, project managers	Commit to duration for individual project CCPM plans and first buffer reports	Plan
9	Leaders	Commit to plan all future projects using CCPM	Commitment
10	Leaders	Determine project priority (or sequence for the drum resource)	Project priority list
11	Leaders	Assign responsibility to create the CCPM plans	Individual critical-chain project plans
12	Leaders	Decide on the CCPM schedule tool	Schedule tool, procedure
13	Leaders	Identify who requires what training	Training matrix
14	Top leader	Commit to formally announce CCPM (duration)	Commitment letter, e-mail, or meeting
15	Project and resource managers	Commit to communicate CCPM to people (duration)	Individual communication
16	Drum resource manager	Commit to building the drum schedule (duration)	Drum schedule
17	Project managers	Commit to weekly buffer meetings	Weekly buffer meetings
18	Facilitator	Get commitment for follow-up session	Follow-up session

implementation meeting, as people forget training very rapidly if it is not immediately reinforced by application in the field.

Table 9.4 lists the steps necessary to begin implementation. Your leadership session may have identified the need for additional training. Deliver this training on an as-needed basis, and do not let it delay proceeding with steps 4 and 5 of the process. You must perform steps 3, 4, and 5 with each individual project team. These

Table 9.4 Implement Phase 2: Individual Project Critical-Chain Plans

Step	*Resp.*	*Action*	*Output*
1	Facilitator	Deliver two-day workshops	Knowledge
2	Trainer	Train software users (if necessary)	Software skill
3	Project managers	Verify or create individual project plans suitable for critical-chain plans, including normal (low-risk) task-duration estimates	Individual project critical-path plans
4	Resources	Determine average task durations	Input data to create plan (including buffer sizing)
5	Project managers	Create the individual critical-chain plans, including sizing all buffers	Individual project critical-chain plans (start dates not yet staggered)

planning sessions should not take more than a few weeks in total. If there are many projects, you may need to have multiple facilitators so that this step does not drag out.

Table 9.5 lists the steps necessary to complete the implementation phase and move into full-scale operation with the CCPM process. Buffer reporting should be initiated within three weeks of the initial leadership training, or the project will flounder. People will begin to forget what they learned about critical chain, and implementation success chances will dwindle.

Once you have moved into initial implementation, you will find a host of items that require clarification and issues that require resolution. You need an ongoing process to ensure that questions are answered promptly, that answers are communicated to all team members with a need (or desire) to know, and that you promptly resolve issues. This process can be part of your measurement-and-control process.

9.1.7 Measure-and-Control Implementation

Management and control of the CCPM implementation project provides the system feedback necessary to move your project management system to the new equilibrium state and keep it there. Your team must install a positive feedback loop to cause the change. The daily or weekly buffer meeting is the primary vehicle for this feedback. It is your lever to lift the world.

Additional feedback during implementation should take into account the following:

1. Prioritizing the projects sets a clear basis for decision making.
2. The drum schedule and staggering of project starts using the drum schedule eliminate much of the serious resource contention for the project teams.
3. Buffer management provides a clear decision-making tool to allocate resources between projects.
4. Project resources are expected to work on one project task at a time and are encouraged by management to protect this mode of operation.
5. Project resources are not pulled away for "higher-priority" projects.
6. Project changes and subsequent rework are reduced due to later starts and earlier project completion.
7. Project changes are reduced because the critical chain does not change.
8. Project changes reduce because the buffer-management thresholds for action are much wider than tolerances usually placed on project-performance variation.

Table 9.5 Implement Phase 3: Drum Schedule and Project Schedules

Step	*Resp.*	*Action*	*Output*
1	Drum manager	Create initial drum schedule	Drum schedule
2	Project managers	Schedule individual projects	Project schedules
3	Trainer	Train resources in relay-racer behavior and using buffer report to set their individual work priorities	Knowledge
4	All project team members	Initiate task priority lists and RDU (remaining duration)estimate	Buffer reports and action plans

Management can enhance the effect of these natural feedback results by assuring communication throughout the project-performance system.

There are many natural feedback loops that will help keep the CCPM system stable in its new state. These include the following:

1. Workers experience less stress (a positive feedback) when multitasking is removed.
2. Project teams experience positive feedback from successfully completing projects.
3. Management experiences positive feedback for increased project success.
4. Management experiences positive feedback for increased profitability.

While some of these higher-level feedback results start as soon as the projects begin to perform to the critical-chain plan and management begins to model the new behaviors, the feedback is relatively weak until the projects begin to complete.

Once management has planned and sequenced the projects, their primary roles are to

1. Participate in the buffer-management process;
2. Ensure that any new projects posed for inclusion into the system are prioritized and fit into the drum schedule.

9.1.8 What if Implementation Progress Stalls?

Sometimes implementation projects stall. This sometimes occurs near the beginning of the project but can occur at any time. People attend the training sessions and meetings and create the work plan. They then seem to completely forget everything in the cold light of the next Monday morning. They immediately drop back into the behavior patterns demanded and rewarded by the present system. Symptoms include complaints such as, "We are too busy to do critical chain," or "The software or procedures aren't working right," or "The managers are not 'walking their talk'." You can expect one consistent symptom: *no one blames himself or herself*.

If you have been working on the implementation for more than three months and are not essentially there and beginning to get positive feedback, you are stuck. You need to dig into your organization's policies, measurements, and behavior to find out where you are stuck and implement a remedy to remove the block. I cannot give you a generic solution because all organizations are different. However, based on W. Edwards Deming's assertions that 96% of organizations' problems are caused by management (seemingly confirmed by my own observations), I can suggest that you start by looking at your leadership.

9.2 Organization Change Theory

Change-management theories abound. Much of the literature starts with the reasons change attempts fail. Failure means that planned change does not achieve some desired goal. Change goes on in all organizations. It just isn't the change wanted by the book authors or the people they asked. The theories rarely agree on the reasons

changes fail to achieve the goal. This should give us pause to suspect that the authors are missing a deeper systemic cause.

People approach process change with caution. The record of successful change is not good. Murray Dalziel and Stephen Schoonover [2] note, "Technocratic leaders, on the other hand, focus exclusively on outcomes without considering the concerns of employees who must implement and sustain change." Many project managers are technocratic leaders (myself included) and, therefore, are subject to this blind spot. Dalziel and Schoonover further note, "This perspective frequently results in short-term gains, unforeseen pitfalls, and long-term resentments."

The technical aspects of CCPM are not challenging to any organization with basic project-management capability. Many organizations with rudimentary project-management capability have been able to use critical-chain implementation as the focus to improve overall project success. CCPM is not so much an advanced project-management method as it is a better and different method.

9.2.1 Seven S Model

CCPM is a management-change system. Implementation must address all aspects of the system. Figure 9.4 illustrates the Seven S model (which I understand to have been developed by McKinzie but modified by many authors since). Dr. Stephen Covey [3] modifies the model to further emphasize the people part of it, and calls it the PS Paradigm. The "P" stands for *people*. These models provide a system definition for change.

Consider the Seven S model for your organization as you plan implementing critical chain. You mostly must assure that your change plan does not overlook some facts about your organization that may block implementation or cause unintended consequences. There is no reason to change the structure of your organization to implement CCPM. Organizations ranging from full project to full matrix have successfully implemented CCPM.

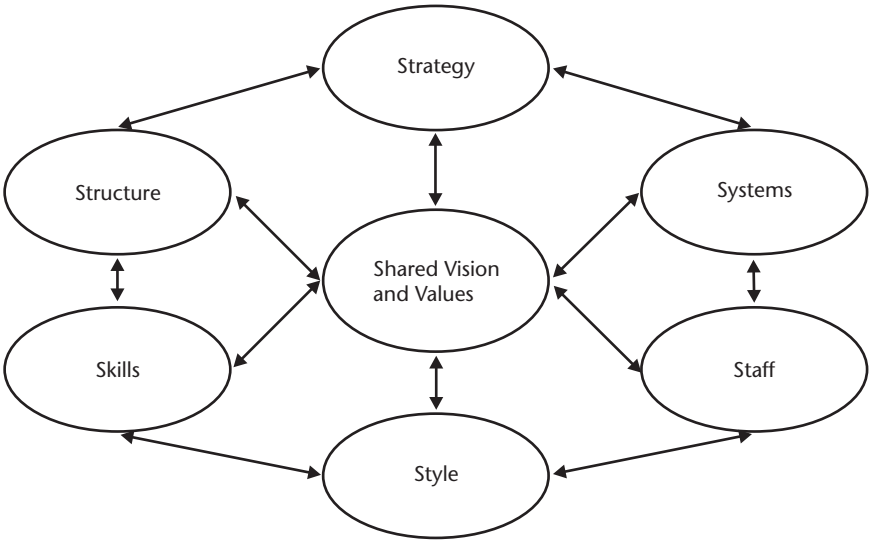


Figure 9.4 The Seven S model provides a system definition for change planning.

The Seven S model is deliberately, as with all models incomplete. Deming emphasized the need for managers to consider elements normally considered outside the business system, such as customers, suppliers, and even competitors. Further, the Seven S model is static. All business organizations are complex, dynamic, and adaptive systems. These means that they are constantly changing. Management's problem is to lead changes in a positive direction.

The process to achieve any goal requires effective feedback mechanisms to adjust actual performance to plan. Buffer management works to determine if you are accomplishing the implementation project according to schedule, as in any project. Also, as in any project, you need result-quality measurements to assure that the results achieved and passed on from one task to the next satisfy deliverable requirements.

Most organizations require behavior changes to implement CCPM. Which changes your organization requires depend on the behaviors your organization currently exhibits. Table 9.1 summarizes how you might assess your organization vs. the desired behavior considering the seven S's for your organization and some typical behaviors.

The following features of CCPM make it easier to implement than many changes people attempt to make in organizations:

1. Nobody loses.
2. Many win.
3. It is simple.
4. Results feedback very rapidly.

Unfortunately, not everyone will feel enough need to change, and others may fear that they have something to lose. The following explains some strategies to deal with these potential obstacles.

9.2.2 3–4–3

The following anecdotal observations are examples of the parables that form the basis for much management theory. A speaker at a management conference brought up the rule of "3–4–3." He said he learned it from a Japanese colleague, as they were studying the application of *Kaizen* (the Japanese word for continuous improvement), the subject of his talk. The reference means that in an attempt to introduce new ideas into any organization, about three out of ten people will catch on immediately and begin to implement. Another three out of ten will remain clueless about and uninterested in just about anything and everything, forever. The middle four of the ten will behave in exactly the way you might expect middle-of-the-roaders to act; they will wait and see, and gradually come aboard as the change becomes the organizational norm.

The time for organizational change to take place depends, of course, on both the change and the organization. For example, I read that it took the British army 125 years to accept the recommendation that foot soldiers should wear asymmetric shoes, that is, one for the left foot and one for the right. Can't you just hear the supply sergeants, "Hey, you guys want to me to be efficient. And now you want me to double my inventory! And think of how hard it is going to be to keep the left and the

right shoe of the same size together, all the way from manufacturing, through distribution and supply, to the soldiers. Costs will go up. Delivery will be harder to maintain. And the soldiers will never be able to put them on; you have made things twice as complicated!" This argument, or something like it, must have held out for those 125 years. (The British may not talk exactly like that, but the gist was no doubt the same.)

In science, it is now generally accepted that it takes at least a generation (i.e., 25 years) for a new, basic scientific theory to replace the old. (Some suggest at least two generations.) The believers in the old theory have to die before the new theory can replace it. This, of course, is for our new and enlightened age. They used to put people like Galileo in jail. Before that, it was "off with his head." You can understand why the scientific revolution took such a long time to build up steam.

Warren G. Bennis [4] identifies three groups of change strategies:

1. Empirical and rational strategies, assuming that people will follow their rational self-interest;
2. Normative reeducative strategies, assuming that change requires alterations in organizational structure, institutional roles, and institutional relationships;
3. Power strategies, requiring compliance with the leaders will.

He describes eight types of change programs that derive from these strategies:

1. Exposition and propagation;
2. Elite corps;
3. Human-relations training;
4. Staff;
5. Scholarly consultation;
6. Circulation of ideas;
7. Developmental research;
8. Action research.

All seek to use knowledge to gain some desirable end. Most of the strategies rely on rationality. Bennis notes that knowledge about something does not automatically lead to effective action. He concludes by observing,

> Sometimes, the changes brought about simply "fade out," because there are no carefully worked out procedures to ensure coordination with other interacting parts or the system. In other cases, the changes have "backfired" and have to be terminated because of their conflict with interactive units. In any case, a good deal more has to be learned about the interlocking and stabilizing changes so that the total system is affected.

Business change advocates in recent years often focus on the need for organizational culture change to precede significant performance improvement. They suggest that it takes many years, perhaps eight to twelve, to accomplish significant culture change in organizations. That's one of the reasons a lot of businesses go out of business before they can accomplish the necessary change.

None of these descriptions, while possibly completely correct, reaches even the correlation level of scientific thinking. For every anecdote someone describes, someone can provide an alternative story that is the exception that proves the rule. Many companies grow to a very large size within the shortest time frame for organization change (eight to ten years.) Others hardly change at all for periods longer than the longest time given for organizational change (e.g., the catholic church.)

9.2.3 Appreciation for a System

Considering an organization as a dynamic system moves our thinking beyond correlation, into the realm of scientific thinking. You can use a model of the present system to aid determining what changes will impact the system the way you want. Dynamic models are important because business systems are dynamic. The Laws of the Fifth Discipline discussed in Chapter 2 apply. One of the most important and difficult to appreciate laws is that causes and effects are displaced in time and space. This means that the effect you observe in Milwaukee today may be due to some management action taken in Tampa last year, not due to the new manager that just came aboard in Milwaukee. The new manager simply *correlates* in time and space with the effect you are observing. No one seriously believes the outcome of the Superbowl causes the stock market to do anything, but every year the media discusses remarkable correlations.

Correlation of effects in dynamic systems makes causes very difficult to determine and often difficult to describe. The definition of cause and effect is that the effect invariably follows when the cause is present. (The effect may also be present without the cause in question if it can also follow from additional causes.) The cause of effects in dynamic systems most often is the system structure, not a specific event. Most people have difficulty gaining an intuitive appreciation for this.

Consider chickens and eggs. The question of which came first is meaningless in a dynamic system that includes chickens and eggs. They coexist. Their numbers correlate in time; that is, everything else being equal, the more chickens you have, the more eggs you get. It is true that chickens cause eggs. It is true that eggs cause chickens. Thus, entity causality is completely circular. This is fine in a dynamic system. Based on the cliché, it does not appear to be intuitive. Depending on the system, it may or may not follow that the more eggs you get, the more chickens you get. Someone may be eating a lot of eggs. This would be part of the system structure and would significantly affect the number of chickens over time.

The Thinking Process that Dr. Eliyahu Goldratt recommends models reality with a cause-and-effect tree—the CRT. This tree structure allows for the inclusion of nonlinear effects and dynamic feedback. It does not provide a way to test and understand the relative importance of system entities and relationships dynamically, but treatment of the feedback loops (more and more entities in the model) attempt to provide a qualitative understanding of the impact. The method helps plan a move from current reality to a desired model of future reality (the FRT) by identifying a core conflict. The core conflict plays a central role in many of the current system UDEs. This conflict is one of the driving forces that maintain the system in equilibrium. Goldratt's Thinking Process always provides a starting point for planning change and usually selects one of the more influential parts of the system to begin the change. Some system thinkers identify the influential parts of the system as "leverage

points." They usually involve a feedback loop. The most effective feedback loops in organizations involve the performance-measurement and reward systems.

The core conflict often involves a measurement or policy of the system. It always influences the behavior of people in the system. The Thinking Process method then surfaces underlying assumptions to identify a starting point to modify the system. Eventually, the process looks to install feedback loops into future reality in order to accelerate movement to future reality and to maintain the system in the new equilibrium.

General system theory and system dynamics teach us that feedback loops are one of the most important elements of understanding and influencing system behavior over time. Feedback loops are the forces that maintain the system in equilibrium and can be used to drive the system to new equilibrium. Measurement systems comprise the primary feedback loops that drive business-system behavior.

9.2.4 Resistance to Change

Resistance to change is an *essential* feature of any stable system. Open systems are only temporarily stable because the dynamic forces acting on the system, both internal and external, are nearly in balance. Please read the first sentence of this paragraph three more times.

Resistance to change is not inherently good or bad. You may judge resistance to change as good if you wish to maintain certain characteristics of a system. For example, you may be very pleased that your system maintains a focus on customer service through good times and bad. On the other hand, you may judge resistance to change as bad if you are attempting to eliminate undesirable behavior or to move to new levels of performance. Regardless of your judgment on the matter, the system will naturally resist change.

Figure 9.5 illustrates just a few of the interrelated forces that exist in any business system. Forces are both internal and external to the business system. The forces are, themselves, interrelated in a complex system structure. Attempts to change any part of the system impact all parts of the system to varying degrees. Because of the linked structure, the net result of these forces will tend to restore the system to its present state.

Resistance to change of the organizational system is often difficult to distinguish from individual resistance to change. When things are not going as hoped, or not quickly enough, people often want to search out and motivate the guilty parties. Unfortunately, such searches are fruitless. How many people do you know who have really wanted to lose weight, quit smoking, or change some other personal behavior, but seemed unable to do it? Or, if they were able to do it, were unable to sustain the progress they made? Do you really doubt their desire or motivation to make the change? Do they not have the skills? Will haranguing them more cause it to happen? Should you send them to training?

The obstacle to organizations making change is the very thing that makes them what they are in the first place. The structure of the system determines the reaction that will happen when you try to push a stable system in one direction. You will activate the restraining forces that helped keep the system in balance where it was. You have to consider resistance to change at both the individual and the organization levels.

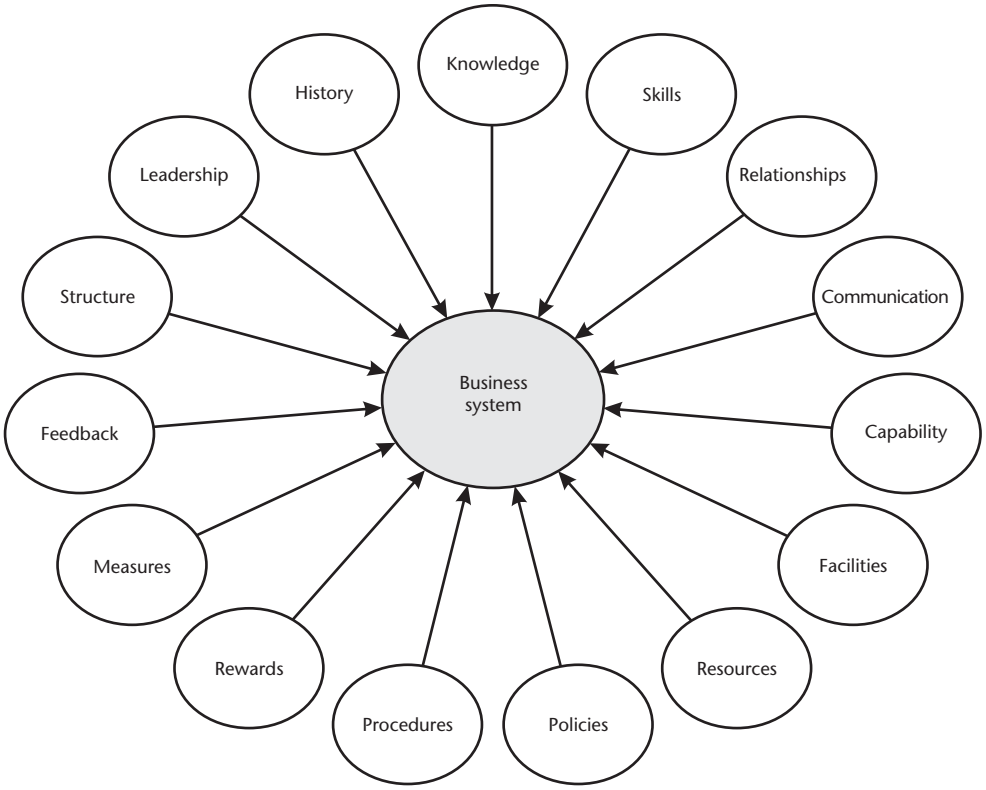


Figure 9.5 Business systems exist in a field of interrelated forces, which naturally push back on attempts to change the system.

For example, consider an organization wishing to become more efficient. It may choose to eliminate excess resources. TOC teaches that an efficient system can only maintain the constraint at full efficiency. All other resources must have protective capacity to operate the system efficiently. In other words, all other resources *must* operate at lower efficiency so that the system can operate at maximum efficiency. Unless the company has a good grounding in TOC, it will not understand the necessary protective capacity and will cut into necessary capacity. This will make the system less efficient. The system will resist the improperly imposed attempt to change it. In some cases, due to some of the laws of system dynamics, the system may appear to be more efficient for a few quarters. This is because there was excess inventory in the system, which can make up for the haphazard cutting of capacity. Once this is used up, the system will begin to fail.

9.2.5 Paradigm Lock

A paradigm is a belief. People often hold beliefs at such a fundamental level that they do not even realize that they hold the belief. Hidden paradigms or beliefs may, nevertheless, influence our thinking. Consider a simple one that many people pay little attention to. The normal practice in critical-path project planning is to create the critical-path schedule and then perform resource leveling. Most people accept this practice without question. They believe it to be acceptable, since they were taught it as the method to plan projects. It seems logical and harmless. They do not think to question it. After all, if you spend the time to question everything, you won't have

time to learn anything. Had they done so, however, they might have asked the following questions:

1. Why does a critical-path plan only connect tasks in a project schedule by output to input to start with? The tasks need both the predecessor input and the resource to begin the task. They are equally important.
2. What happens to the critical path after a project plan is resource-leveled? Does it include the same tasks as before? The previous critical path through these tasks now has slack because time was added to other paths by giving the resource to the critical-path task first. The definition of the critical path is the path with no slack.

Many people recognize problem 2. People find that resource leveling significantly extends the project schedule, sometimes beyond the due date. Therefore, they choose to go back to the nonleveled schedule with the thought they can somehow make it work. They then identify one of the causes of project failure as insufficient resources.

A paradigm shift is a change in our beliefs. Paradigm shifts happen very suddenly. People shift a belief from one pattern to another in an instant. Unfortunately, the shifts do not happen with ease. People are often stuck in a "paradigm lock." They can understand the benefits of changing their behavior, and perhaps even their beliefs, but somehow they just can't do it. Project managers and resources *know* that the duration people put into schedules are highly uncertain. It is illogical to assign start and stop dates to hundreds or thousands of these tasks linked together in a schedule. Nevertheless, they just can't buy the idea that you can live without doing this. They are locked into the deterministic-date paradigm.

Samuel Culbert [6] addresses paradigm lock as a major cause of apparent resistance to change, noting, "What is fundamentally necessary is that you understand that distinct interests and motives exist and are the driving force behind people's participation and that these are neither known to you nor under your control." In other words, you may not have a general case of paradigm lock; you may have many cases with different paradigms. He describes the hypothesis underlying his thesis as the artifact of mind insight: "Organization Is an Artifact of The Mind That Views It." As a direct consequence, he suggests, "the people who are targets of the advice are less inclined to experience advice as valuable counsel and more inclined to see it as resistance and self-interested and agenda-biased opposition to how they want to proceed." In other words, you are resisting, not them! Culbert concludes, "there is no formula for producing this type of change, no matter how vivid and compelling the data and life situations you've got at hand." In other words, there is no repeatable solution to paradigm lock.

9.3 Goldratt's Resistance Model

Goldratt developed a model he calls the "six layers of resistance" to describe the personal aspects of resistance to change. Although I have not found the model especially useful to plan and execute change, it makes a compelling story and can open up communication about the change process. This model supplements and does not

replace the system analysis and behavioral approach recommended above. Dr Goldratt's six layers of resistance are defined somewhat as follows (he sometimes changes it a little bit):

1. Not my problem! (When confronted with a problem, people frequently tend to blame others for the problem and at least to disclaim responsibility or accountability for correcting the problem);
2. Not in agreement with the direction of the solution (people tend to want to stay within existing patterns, thinking if the thingee didn't work last time, it must be because we didn't use it right, so let's try harder next time);
3. Not in agreement with the specifics of the solution;
4. Yes, but . . . it will cause some unintended negative consequence (e.g., management or the customer will take away our buffers);
5. Yes, but . . . it'll never work here (there are obstacles to implementation that they believe to be unique to their environment);
6. Unspecified fear that prevents moving ahead (paradigm lock).

Goldratt states that people usually traverse this model in the sequence listed. If they get partway along and feel stuck or lose understanding, they tend to drop all the way back to layer one, rather than just remaining stuck on the higher-level layer of resistance.

The 3–4–3 model may generally describe how intensely people will experience these layers of resistance.

9.4 To Pilot or Not to Pilot?

The most common response offered by organizations considering changing to CCPM is, "Let's try a pilot project." I used to discourage this approach. My discouragement has never worked. So, when you plan a pilot project, please consider the following:

1. First and foremost, pick a project led by someone who really wants to deploy CCPM.
2. Specify, in writing, clear expectations for the pilot project.
3. Prepare a charter for the pilot project.
4. To the extent possible, pick a project in which most of the resources can be shielded from multitasking demands by other projects.
5. Train the entire project team before the project.
6. Involve the project team in creating the CCPM schedule and ensure that they take ownership in the schedule.
7. Have all software and operating procedures in place prior to project kickoff.
8. Ensure good project-management processes are in place and used in the pilot (e.g., change management, risk management, issue and action management, and communication).
9. Monitor the health of the project team frequently and work with them to resolve issues that arise.

If you follow these guidelines, you should have a successful pilot project and be able to communicate the short-term wins that the pilot project brings you.

9.5 Example Objections

Listed below are some of the objections I hear while implementing CCPM. I do not hear all of them in all organizations, and there are many more voiced in some organizations. This list is just to give you an idea of what to expect. You can think of it as a David Letterman top-ten list (but I've provided a few more for your enjoyment and contemplation). Do not lose heart when you hear them: all of the organizations I have heard them in ultimately did fine. I do not provide answers as by now you should know what the answers are, and your answers for your organization will probably differ from mine. The best way to deal with these is to listen, acknowledge, and simply ask the person who stated it to give it their best shot.

1. If this is so good, why haven't all the other companies in our business implemented it?
2. We already use overly optimistic estimates and always work to meet or beat them.
3. It'll never work here. CCPM is doomed to fail given the inherent complexity of our work.
4. It'll never work here. Our projects have too much uncertainty.
5. Management will never prioritize projects.
6. People early in the project will steal all the buffer.
7. Management will beat us up when our buffers are in the red.
8. CCPM won't work because of all the nonproject work we are expected to do.
9. Management will not change, so why should I? Do you really expect me to believe that clear, unambiguous priority calls will be made across our portfolio?
10. CCPM doesn't fit with our historical focus on milestone performance.
11. I seriously doubt we are capable of sequencing work into the system based on an assessment of our resourcing capacity.
12. People in my department/section all have varying levels of skill and experience. One cannot automatically replace another in a CCPM environment.
13. I doubt others are not going to hold me accountable for due dates at the task level. If they don't, how am I being held accountable?
14. I get rewarded for multitasking, in both good and bad ways!
15. It will never work here; there are way too many obstacles operationally and culturally in our environment. We're wasting our time. Let's simply get on with the real work.
16. What's so bad about the way we currently manage projects? If it's not broken, why are we trying to fix it?

And lastly, my personal favorite:

17. We can't implement something that will show a 50% to 100% improvement. It would mean I have done a bad job my whole career!

9.6 Key Points

This chapter has provided the outline of a plan for the change to CCPM in a multiproject environment and the supporting theory. You can implement it on a single project with a simpler plan. Key points presented in this chapter include the following:

- Effective change plans harness organizational dynamics to accelerate the change.
- Senior-management leadership is the critical success factor for multiproject CCPM implementation.
- Use your project process to implement the change to CCPM: charter, endorse, work plan, perform, and close.
- Carefully define your pilot project(s), and support them.
- Humans and organizations try to maintain equilibrium (i.e., appear to resist change). Your implementation plan should anticipate and plan for this.
- *Just do it!*

Organizational understanding of TOC can greatly aid implementing CCPM but does not seem to be a necessary condition. Your organization's history of change should provide you with clues as to how hard a transition you might have ahead of you and help you plan for it. You can make substantial improvements on your individual projects using the CCPM principles, even if your organization does not choose to move to a multiproject implementation in the beginning.

References

[1] Kotter, J. "Leading Change: Why Transformation Efforts Fail." In Harvard Business Review on Change, pp. 1–20, Boston, MA: Harvard Business Review Publishing, 1998.

[2] Dalziel, Murray M., and Stephen C. Schoonover, Changing Ways, A Practical Tool for Implementing Change within Organizations, New York: Amacom, 1988.

[3] Covey, Stephen R., Principle Centered Leadership . . . Teaching People How to Fish, Provo, UT: The Institute for Principle Centered Leadership, 1990.

[4] Bennis, Warren G., Kenneth D. Benne, and Robert Chin, The Planning of Change, New York: Holt, Rinehart and Winston, Inc., 1969.

[5] Skinner, B.F., Science and Human Behavior, London: The Free Press, Collier Macmillan, 1953.

[6] Culbert, Samuel A., Mind-Set Management, The Heart of Leadership, New York: Oxford University Press, 1996.

CHAPTER 10

Project-Risk Management

Project-risk management seeks to manage and control the risk of project success to an acceptable level. Project risk deals with the risk to project success in terms of scope, cost, and schedule, including customer satisfaction. Other processes deal with other risks, such as health and safety risks or environmental risk. Project-risk management seeks to control project risks beyond the scope of your project plan and beyond your "circle of control."

Project-risk management is part of the project-planning process because you must decide on the course of action to include in your project plan based on the relative risk. Whenever you make a project assumption, you are making a project-risk decision because you are assuming reality in the future will follow your assumption. If your assumption does not come true, you have a project-risk event.

The PMBOK™ Guide [1] suggests that risk management can include opportunities as well as negative consequences. If your project environment has significant upside potentials that you are unable to include in the project baseline and there may be benefit to addressing them (e.g., you might be able to influence the probability or consequence), you may wish to use the following process for the upside consequences as well. If you choose to do so, I recommend using a separate table for positive risks. Please make the mental changes to what follows if you are considering positive risks.

Project managers have several options to deal with project-risk events, including

1. Spending effort to prevent the occurrence of the risk (e.g., limiting the use of flammable materials to prevent a potential fire);
2. Identifying and monitoring the risk triggers (e.g., reviewing weather forecasts and monitors);
3. Taking preventive actions that may reduce the potential consequences of the risk, should the event occur (e.g., spill-control dikes);
4. Purchasing insurance;
5. Planning for mitigation in case a risk event occurs (e.g., fire department);
6. Accepting the risk.

Critical chain simplifies conventional project-risk management because it need only deal with special-cause risks. The CCPM process provides the necessary and sufficient process and tools to deal with common-cause risks to schedule and cost, and to scope to some degree. The project-quality process is also a risk-management tool, protecting the project from scope risk.

The PMBOK™ Guide [1], its more-detailed support information [2], and much of the project literature (e.g., [1–4]) fails to discriminate between common-cause variation and special-cause variation when addressing project risk. We note in Section 2.5.2 that Dr. W. Edwards Deming, the father of TQM, described this as a fatal error [5].

10.1 Defining Project-Risk Management

Risk has two components: the probability of a risk event and its impact on the project. We can loosely define risk as the product of multiplying these two components.

Risk types include

- Program risk: These may cause client dissatisfaction and include the risk that the client need is not known, that the full scope to fill the need is not known, or that project assumptions may not come true.
- Business risks: These may affect the impact the project will have on the rest of the business; they include financial risks and risks to the company's reputation.
- Technical risk: These are risks of developing or applying new technologies that are not in common use. This includes, for example, a new drug development discovering an unanticipated side effect.
- Cost risks: These may impact the project beyond one third of the project's cost buffer.
- Schedule risks: These may impact the project beyond one third of the project's schedule buffer or beyond a feeding buffer.
- Health and safety risks: These have the potential to injure the project team or public beyond the risks routinely accepted by the public.
- Environmental risks: These may impact necessary project conditions (scope, schedule, cost) as a consequence of some environmental variable.
- Regulatory risks: These may impact necessary project conditions (scope, schedule, cost) as a consequence of some regulatory impact, such as a new design requirement or constraint or delay.

10.2 Risk-Management Process

Figure 10.1 illustrates the project-risk-management process. It starts with identifying the risks your project may encounter.

Risk assessment may be quantitative or qualitative. Quantitative risk-assessment tools include failure modes and effects analysis, Monte Carlo Analysis, project simulation, PERT, probabilistic safety assessments, and the Management Oversight or Risk Tree. For risks reducible to a quantity (e.g., a cost or schedule days), you can compute a probable impact as the multiplication of the probability and the risk. For example, a 50% probability of overrunning by $100,000 has a probable risk of $50,000. This risk computation is useful as a relative ranking but is

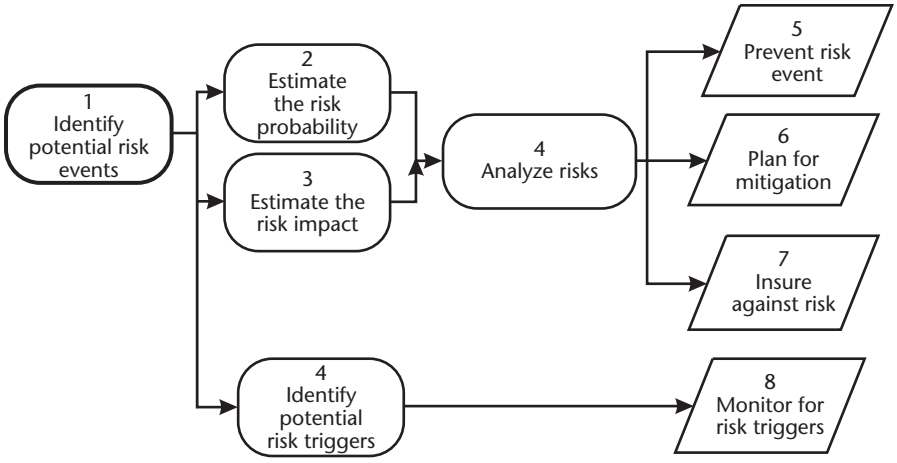


Figure 10.1 The project-risk-management process.

only quantitatively useful if you can insure against the risk. Otherwise, it is going to cost you nothing or $100,000, never $50,000.

I focus on qualitative risk assessment with risk ranking because the data is usually not available to justify detailed quantitative risk assessment, and supplying probable risk numbers tends to yield a false sense of believability.

10.2.1 The Risk Matrix

Table 10.1 illustrates the basic risk-management matrix. It summarizes the risk, the assessment of the risk, and the planned actions to monitor, prevent, or mitigate the results of the risk. The content in the table is only for illustration; your content should be much more specific to your project. However, I do encourage you to follow the lead of combining like risks in order to keep the overall length of the list to a reasonable number of items (i.e., less than a dozen or so). You should define what a reasonable number of items is based on the overall risk and size of your project. Relatively small projects (i.e., less than a few million dollars and under one year) should not have a risk list in excess of ten items. If your list for a project of this size just seems to have many more high-impact, high-consequence risks, you should ask yourself if you really want to do that project!

The second column in Table 10.1 gives a description of the potential risk event. You may start with a list of many specific events that people can imagine and then lump them together for subsequent analysis. You may also categorize risks in terms of their probability and potential impact; the next two columns; that is, you may have one event for low-impact natural events and another for high-impact natural events. The reason to do this is that the two types of events will lead to different mitigation strategies.

David Hilson [6] provides a risk-statement format that I find useful to clarify the risk: "As a result of <cause>,<effect> may occur, which would lead to consequence." I have used italics for the effect in the example risk statements in Table 10.1 to illustrate how to use the format.

Columns 3, 4, and 5 provide a relative quantification of risk. Risk is the product of consequence and probability. The legend below the table provides one possible

Table 10.1 The Risk Matrix

#	*Risk Event*	P	C	R	*Trigger to Monitor*	*Prevention Actions*	*Mitigation Actions*
1	As a result of a **natural event**, *work delay* may result, which would lead to delaying project completion	1	3	3	Weather reports Trends	Plan outside work during dry season	Erect tents and tarps over work areas Build dikes around facility Implement a high-wind design Implement a seismic design Install pumps for rain and flood
2	As a result of **fire**, *loss of supplies or building* may occur, leading to **loss of entire project**	1	3	3	Fire prevention inspections Alarm system	Use noncombustibles Fireproof storage cabinets	Install fire suppression ASAP
3	As a result of being *unable to meet technical development objectives*, technical performance may not be achievable, leading to **inferior product**	2	2	4	Development tests and gates	Develop quality process and parallel alternatives	Develop an alternative technology
4	As a result of **regulatory impact**, *additional work may be required*, increasing cost and/or **delaying schedule**	2	2	4	Excessive questions or no action from regulators	Hold face-to-face discussions with regulators Use consultants to prepare applications	Put together a task team to respond to causes of delay or denial
5	As a result of **supplier delay**, *we may be unable to build components on time*, **delaying the overall project**	3	2	6	Late contracts Delayed delivery	Impose late-delivery penalties Buffer deliveries Check supplier delivery references	Prepare alternative suppliers and equipment

Legend:

P (Probability)
3 = 20–50%
2 = 5–20%
1 = <%%

C (Consequence)
3 = >Project Buffer
2 = ≥20% of Project Buffer, ≤Project buffer
1 = <20%

$R = P * C$

way of quantifying the probability and consequence of a risk for a project. Note that the probability refers to the probability of the risk occurring during the project. This probability only goes to 50% because if you judge the probability to be higher than that, you should assume that the risk event will occur when preparing your project plan; that is, risks with a probability greater than 50% should be treated as a baseline assumption. The impact is put in terms of the project buffer for schedule and may be put in terms of the cost buffer for cost. You may have additional consequence criteria, such as safety risk or public-reaction risk. Section 10.3 provides additional detail on qualifying and quantifying risk. Note that with the recommended scale, risk quantification can range from 1 to 9.

The sixth column in Table 10.1 lists the triggers to monitor. You should assess these frequently to see if you should change your risk assessment or activate your contingency plans. You should, of course, attempt to come up with leading indictors whenever possible.

Columns 7 and 8 of Table 10.1 are the most important as they list the actions you will take to prevent or mitigate the potential risk. Prevention and mitigation may work on either the event probability or the event impact. For example, a spill-control dike reduces the potential impact of a spill but not the probability. On the other hand, a double-wall tank reduces the probability of a spill. Actions to prevent the risk should then become part of your project work plan. Actions to mitigate may require actions in your project work plan to plan for mitigation, such as training or purchasing emergency supplies.

10.2.2 Incorporating Risk Assessment into the Project Process

Your risk assessment is only as valuable as what you do with it. Listing risks might give you ammunition to say, "I told you so". It also opens you to the question, "why didn't you do anything about it?" You must take action on the identified risks to have any result from your risk analysis. Your actions might include

- Preventing (or reduce the likelihood of) the risk event (e.g., breaking the project into phases or researching uncertain project elements to improve certainty);
- Transferring the risk (e.g., subcontracting);
- Monitoring to determine if the chances of the risk are increasing (e.g., monitoring for precursors);
- Reducing the consequences of the risk event, should the risk materialize;
- Insuring against the risk event;
- Mitigating the consequences if the event the risk materializes.

You may choose to use a consistent approach to applying these alternatives, such as illustrated by Table 10.2.

Table 10.2 Guideline for Processing Potential Risk Events

		Probability of Risk		
		High (3)	*Medium (2)*	*Low (1)*
Consequence of risk	*High (3)*	Prevent event Reduce consequences Plan to mitigate Monitor	Plan to mitigate Monitor	Plan to mitigate Monitor
	Medium (2)	Prevent Plan to mitigate Monitor	Plan to mitigate Monitor	Monitor
	Low (1)	Monitor	Monitor	Ignore

10.3 Identifying Risks

10.3.1 Risk List

You may use a variety of methods to identify risks. One method starts with the assumptions your team felt necessary in order to develop project-work estimates. Each of these assumptions represents a risk of not being true. You may use checklists, such as the one included in R. Max Wideman's Appendix A [2]. You may use computer assistants, such as included in some software offerings [7]. You may simply get your project team together and brainstorm a list to start with (this is the approach I usually take). You may evaluate the problems encountered by previous similar projects. Coming up with the list is usually the easy part. You will never be able to predict the future, so you will never be able to come up with a complete project-risk list. It would be infinitely long anyway and not very useful to your team. Instead, you should seek to obtain a representative list of the type of risks likely to confront your specific project during its time of execution.

10.3.1.1 Project Assumptions

Many of your project assumptions may translate to project risks if the assumptions do not come true. For example, your assumption that regulatory permit reviews will take 60 days low-risk and 30 days average duration may become a risk if the reviews take longer than two thirds of your project buffer. You may have reason to expect that the reviews could take much longer based on recent experience with the same regulatory agency on another project.

On the other hand, you should guard against too many project assumptions. You need to ensure that a rule of reason applies in specifying both the project assumptions and the associated risks.

10.3.1.2 CheckLists

Checklists often help to identify risks you might otherwise overlook. Checklists have two inherent problems:

1. Checklists may suggest risks that are not significant to your project but that become believable once suggested.
2. Checklists may lead to overconfidence that you have considered everything important, limiting your search for things beyond the checklist.

The rule of reason applies.

10.3.1.3 Plan Scrutiny

You should scrutinize your plan, asking what could go wrong in each of the major steps, to aid in developing your risk list. You can let this list get relatively long while preparing it, as you will consolidate it in the next step.

10.3.1.4 Consolidation

If your risk list begins to get long, you should group like items to consolidate your risk list before going on to select the risk actions. Your purpose is to come up with a

reasonable set of risk items to manage. Increasing the detail of the risk does list does not increase its accuracy. There are, in actual fact, an infinite number of potential risk events. You can never list them all. It is far more important that you capture the important types of risks and put in place the appropriate information and response system to deal with the actual risks that arise. You lose focus if the list of individual risk events becomes too long; and it is impossible to plan realistic actions to prevent or mitigate the effects of too long a list. You should try to limit the list to, at most, a few tens of items. For most reasonable size projects (i.e., less than ~ $10 million and one or two years in duration), the list should be less than ten items, or you probably should not be doing the project.

10.3.2 Classifying Risk Probability

You must make an estimate of the probability of each risk event actually occurring during the life of your project in order to decide on a rational plan to manage the risks. You do not want to spend a large amount of your project resources guarding against low-probability events. On the other hand, you want to prevent events that are likely to occur, and be prepared to handle some events, even if they are unlikely, if the potential consequence is large enough.

Peter Bernstein [8] notes, "The essence of risk management lies in maximizing the areas where we have some control over the outcome while minimizing the areas where we have absolutely no control over the outcome and the linkage between effect and cause is hidden from us." He goes on to note that insurance is available only when the Law of Large Numbers (see 4th bullet) is observed, that is, where the laws of chance work in favor of the insurance company or gambling institution. The very nature or risk, then, ensures that we are dealing with relatively low-probability events to start with.

People's ability to estimate probability is notoriously poor [9–11]. When considering probability, most people are subject to numerous logical biases and errors. Unfortunately, research demonstrates people are likely to be unjustifiably confident in their erroneous "knowledge." I will just list the common errors here to make you aware of them. Overcoming these biases and errors is the topic of another book.

- Failure to understand how probabilities combine: The probability of two independent events is the product of the probabilities of the individual events. Since probabilities are always numbers less than one, the probability of the two events is always lower than the probability of either single event.
- Failure to consider the base rate: The base-rate error fails to consider the distribution in the population. For example, consider a bead drawn from a population of 90% white beads and the probability of correctly identifying a white bead in dim light of 50%. A person looks at a bead under those conditions and says, "black bead." What is the probability that the bead was black? Most people answer 50%. The correct answer is only 5%.
- Availability: The availability error gives unjustified bias toward whatever comes to mind, usually because of a recent reminder but also because it something thought to be "typical."
- Failure to understand the law of large numbers: People routinely accept small samples as indicative of a larger population and fail to understand that the

variance in small samples tends to be much larger than the variance in larger samples of the population.

- Representativeness error: People mistake "more typical" for "more probable." For example, people will claim, based on a description, that a person is more likely to be a school teacher than a working woman because the description included traits people associate with schoolteachers. Since the category "working woman" also includes all women schoolteachers, it is actually more likely that she would be a working woman than a schoolteacher.
- Anchoring: People tend to not deviate much from initial positions put forth by others or themselves, especially in regard to numbers. This bias also allows groups to significantly influence each other. If you want independent input, you have to seek independent input and not have one person review another's work, as a reviewer usually only focuses on what they are given to review, i.e, they are anchored on the review material.
- Confirmation bias: Once people have made a statement or decision, they tend to look for instances that confirm that statement or decision. Unfortunately, confirmatory cases have no value in scientific proof. People should look for instances that would disconfirm their hypotheses. This bias often results in worthless tests. Effective tests must always seek to disconfirm the hypothesis.

You can use this list to critically review your list of risk events and your categorization in terms of probability and impact. Ask, are we making this error?

10.3.2.1 High Probability (3)

You should not have any items on your risk list that exceed a 50% probability of occurring during the life of your project. You should count on items with a greater than 50% probability happening and include them in your project assumptions and baseline plan. You should consider defining high risk as less than a 50% chance of a risk's happening during the life of the project, but as a more than a moderate risk, which you might define as ranging from a 5% to 15% chance.

10.3.2.2 Moderate Probability (2)

The cop-out definition is that moderate-probability risk events are less risky than high-probability events and riskier than low-probability events. They are events that may occur during the life of your project, but you would not bet on them happening (or, at least, you would want very favorable odds on the bet.)

10.3.2.3 Low Probability (1)

Low-probability risks include those unlikely to occur during the life of your project (i.e., they have less than a 5% chance of happening), as well as those with a very low probability (i.e., on the order of 1% or less). Your project design may have to account for risks of lower probability during the life of the project result, such as earthquakes or extreme weather, but that is not the topic of project-risk assessment. Exceptions may include insurance for events such as extreme weather (e.g., hurricane, and flood) on a construction project.

10.3.3 Classifying Risk Impact

You must classify the risk impact to define the risk because risk is the product of probability times impact. You could qualify impact in terms of the overall project schedule and cost or the expected return on investment for the project. CCPM provides a unique measure of classifying the risk impact in terms of the project buffers for time and cost. The buffer size is an indicator of the common-cause risk in the project and, therefore, is a reasonable basis on which to measure special-cause variation.

10.3.3.1 High Impact (3)

High impact is anything that could cause an impact in excess of the project buffer on schedule, in excess of the cost buffer on cost, or otherwise result in client or project-team dissatisfaction.

10.3.3.2 Moderate Impact (2)

Moderate impact is an impact that would consume on the order of two thirds of your project buffers, or all of your feeding buffers, but at least one third of the respective buffers.

10.3.3.3 Low Impact (1)

Low-impact consequences would not exceed one third of your project schedule or cost buffers and not be a significant concern to your client or project team.

10.4 Planning to Control Risks

10.4.1 Risk Monitoring

You should plan on monitoring for the risks you elect to keep in your risk-management list. This means you should, as a minimum, review the list with the team members at preplanned intervals in your project meetings (e.g., once a week or month) and ask if any of the risk triggers seem immanent, if risks are past, or if new risks are perceived. Sometimes you may need more formal monitoring for the risk triggers.

10.4.2 Prevention

Risk-prevention activities you have elected to implement become part of your project plan. All you have to do to ensure that they are in place is to follow through on your project-measurement-and-control process.

10.4.3 Mitigation Planning

Plans for risk mitigation should also be part of your project plan. You should include routine activities necessary to ensure the viability of your risk-mitigation plans, such as fire inspections or emergency drills, as part of your project-monitoring-and-control process. You need not include these periodic or ongoing activities as specific activities in your project network.

10.5 Key Points

Project-risk management controls special-cause variation through monitoring, prevention, mitigation, or insurance. Key points emphasized in this chapter are:

- CCPM simplifies project risk management by eliminating the need to address common-cause variation. CCPM risk management only addresses special causes of variation.
- You must include a risk management process in your project work plan. You should scale the implementation of risk management to overall project risk.
- Project risks must identify the risk event, the probability of the event occurring, and the potential impact or consequence of the risk event to the project.
- CCPM project plans help to define the relative risk in terms of the project buffers.
- The project team must decide among options, including prevention, mitigation, insurance, monitoring, and ignoring risks.

Risk monitoring, prevention, and preparations for mitigation should be part of your project plan. You should assign specific responsibility to monitor risk throughout the performance of the project and update your risk plan as appropriate.

References

[1] PMI, *A Guide to the Project Management Body of Knowledge*, Newtown Square, PA: PMI, 2000.

[2] Wideman, R. Max, *Project and Program Risk Management, A Guide to Managing Project Risk and Opportunities,* Newtown Square, PA: PMI, 1992.

[3] Meredith, Jack R. and Samuel J. Mantel, *Project Management, A Managerial Approach*, New York: John Wiley and Sons, 1985, pp. 68–71.

[4] Wysocki, Robert K., Robert Beck Jr., and David B. Crane, *Effective Project Management*, New York: John Wiley & Sons, 1995.

[5] Deming, W. Edwards, *The New Economics*, Cambridge, MA: MIT Press, 1993.

[6] Hilson, David. "When Is a Risk Not a Risk: Part 2," available at http://www.risk-doctor.com/pdf-briefings/risk-doctor07e.pdf (accessed June 22, 2004).

[7] RiskTrak, Risk Services & Technology, Amherst, NH, 03031.

[8] Bernstein, Peter L., *Against the Gods, The Remarkable Story of Risk,* New York: John Wiley and Sons, 1996.

[9] Kahneman, Daniel, Paul Slovic, and Amos Tversky, *Judgment under Uncertainty: Heuristics and Biases*, Cambridge: Cambridge University Press, 1982.

[10] Belsky, Gary, and Thomas Gilovich, *Why Smart People Make Big Money Mistakes, and How to Correct Them,* New York: Simon & Schuster, 1999.

[11] Russo, J. Edward, and Paul J. H. Schoemaker, *Decision Traps, The Ten Barriers to Brilliant Decision Making, and How to Overcome Them,* New York: Simon & Schuster, 1989.

CHAPTER 11

The Theory of Constraints Thinking Process Applied to Project Management

This chapter integrates the rest of this book and describes the process used to think through project management and develop the critical-chain process. It also provides some ideas for future directions. One of Dr. W. Edwards Deming's obstacles to improvement is a tendency on the part of many managers to search for examples. He states, "My answer to such enquiries (i.e., for examples just like us) is that no number of examples of success or of failure in the improvement of quality and productivity would indicate to the enquirer what success his company would have" [1]. He notes further, "The question is not whether a business is successful, but why? And why was it not more successful? It is necessary to understand the theory of what one wishes to do or make." This text has presented the CCPM theory you need, including some of the supporting TOC, TQM, Six Sigma, Lean, Agile, and PMBOK™ principles. Dr. Eliyahu Goldratt's Thinking Process provides a tool to capture much of that reasoning.

11.1 Synthesizing the Principles

Figure 11.1 illustrates a matrix showing how and where TOC tools can be applied within project management. It also illustrates how TOC thinking complements much of project management, and vice versa. For reasons I don't understand, some have positioned TOC and/or critical chain as "versus" conventional project management. As I indicated in the beginning, I see TOC as one addition to the ongoing improvement of management thinking and appreciate the particular help it provides to project management. This chapter will describe each of the TOC tools indicated in the table and illustrate how they can be used above and beyond the application of the critical chain.

Figure 11.2 illustrates CCPM integration of the major management principles outlined in Chapter 2. The top rows again illustrate that CCPM is the combination of conventional project-management and critical-chain thinking from TOC. The key Lean elements in the table are from J. P. Womack and D. T. Jones [2]; the Six Sigma principles are from P. S. Pande, R. P. Neuman, and R. R. Cavanagh [3]; the TQM principles are Deming's points of Profound Knowledge; and the TOC points listed are the five focusing steps. (Figure 11.1 lists the other relevant TOC tools that will be explored in this chapter.) The main purpose of this figure is to show that the synthesis is multifaceted. There are, or course, relationships between the ideas of

	PMBOK													
	Knowledge Areas									Phases				
	Integration	Scope	Time	Cost	Quality	Human Resource	Communications	Risk	Procurement	Selection	Inception	Planning	Execution	Close-out
TOC Tools														
TOC Accounting				x					x	x				
Five Focusing Steps	x	x	x	x	x				x		x	x	x	
Critical Chain	x		x	x		x	x	x				x	x	x
Thinking Process Tools														
Current Reality Tree	x	x								x				
Evaporating Cloud	x	x				x	x	x		x		x	x	
Future Reality Tree	x	x								x	x			x
Negative Branch					x			x					x	
Prerequistie Tree		x			x						x	x	x	x
Transition Tree					x		x						x	x

Figure 11.1 TOC relates to many areas of the PMBOK™.

each of the disciplines in the columns or rows as well. When a conflict appears to arise between the disciplines, you could do worse than use some of the TOC tools described below to help you resolve the conflict.

The PMI's OPM3 [4] extends the PMBOK™ to the program and portfolio level. Up to this point I have not addressed these levels of project leadership. This is one area you can extend CCPM thinking to. You can use the tools described below or simply start at a high level and ask, how would TOC influence project-portfolio selection? Recent discussion on a TOC Internet group indicates that some simple solutions may apply, but that few have worked out the details yet, and none have published effective approaches. The discussion ended with recommendations to consider the uncertainty inherent in any project projection (uncertainty in the return from the project quite often is far more than the uncertainty of scope, cost, or schedule within the project) and to project the impact on future throughput and operating expense for various scenarios. The discussion rightly highlighted the fact that usually portfolio selections are made in light of previous decisions (i.e., an ongoing set of projects), and that new project proposals should always be judged in light of the ongoing projects, or, if you prefer, they should be judged together.

11.2 Applying Goldratt's Thinking Process to Project Management

The Thinking Process, described in Chapter 2, is Goldratt's attempt to enable the rest of us to create the innovative system solutions that he developed. The process starts with a model of current reality and ends with a plan to create a preferred

	PMBOK Knowledge Areas									CCPM Phases					CCPM Critical Chain Features Single Project					Multi	
LEAN	Integration	Scope	Time	Cost	Quality	Human Resource	Communications	Risk	Procurement	Selection	Inception	Planning	Execution	Close-out	Mean estimates	Critical chain	Buffers	Relay-racer	Buffer management	Project priority	Pipelining
Value				X	X				X	X										X	
Value Stream	X	X	X		X		X	X				X			X	X					X
Flow					X	X	X	X					X		X	X	X	X	X		X
Pull					X			X							X			X	X		X
Perfection					X									X							
Six Sigma																					
Customer focus	X	X					X	X		X		X		X							
Data driven management		X	X	X	X												X		X		X
Process focus	X	X	X	X	X	X	X	X		X	X	X	X	X							X
Proactive management															X			X	X	X	
Boundaryless collaboration	X						X									X					
Perfection					X																
TQM																					
System Focus	X																				X
Understand Variation					X			X							X		X		X		
Understand Psychology						X	X											X	X		
Theory of Knowledge					X									X							
TOC																					
Identify constraint	X					X										X					
Exploit constraint															X		X	X	X	X	X
Subordinate to constraint																			X	X	X
Elevate constraint						X															
Do not let inertia impact					X							X		X							

Figure 11.2 The linkages between CCPM and Lean, Six Sigma, TQM, and TOC.

future reality. It is a logical process that depends heavily on critical review and buy-in to the models.

In seeming contrast to Deming's assertion regarding the search for examples, I stated in Chapter 2 that you do not need to understand TOC tools to successfully apply CCPM. The conflict cloud evaporates by understanding the underlying assumptions of CCPM expressed throughout this book. The understanding of why it is successful may be sufficient for your application. Comparing the assumptions identified in this book to your organization's reality may enable you to adapt the process as necessary. Certain features of the process (e.g., eliminating bad multitasking) appear to be so robust that you are likely to experience positive results with CCPM even if your implementation does not match the textbook representation provided here. Organizations that are relatively flexible (which some call "early adopters") have succeeded in deploying critical chain without knowing the TOC Thinking Process and with only an overview-level knowledge of TOC tools. Other organizations have deployed elements of critical chain under the auspices of some of the other management approaches, such as Lean or Six Sigma.

I had the fortunate experience to see an early version (August 1994) of the Thinking Process applied to project management, developed by Dee Jacob of the Avraham Y. Goldratt Institute (AGI). It included the thinking of Jacob, Goldratt, and others at AGI. It had all the essentials presented in current thinking for single-project critical chain. It made sense to me in view of my years of project experience. The changes from 1994 to the current single-project process are minor in substance and have more to do with displaying the results than they have to do with the essential project-management system.

The initial method did not include the current understanding of multiproject critical chain. Goldratt had devoted only a couple of pages to multiproject resource contention in *Critical Chain* published in 1997 [5]. From my view, Tony Rizzo of Lucent Technologies discovered the significance of the multiproject approach but has not yet published his findings on his own. In that regard, the result seemed to actually lead the process to develop it, i.e., development of the multiproject CRT and FRT followed the definition of the solution.

Unfortunately, few people seem to have the inclination to engage in the deep thinking and scrutiny necessary to follow the Thinking Process. Eric Noreen, Debra Smith, and James Mackey describe some early results with training in the Thinking Process [6]. Their results show poor application after the early Jonah training. (Jonah training is the name that AGI uses for training in the Thinking Process, following the behavior of the character Jonah in Goldratt's first book [7].) My (nonscientific) poll over the last several years and recent discussions on TOC Internet groups lead me to believe that those findings would not change substantially today; that is, few people apply the Thinking Process after the Jonah training, and fewer yet create breakthrough solutions. There are reasons to believe that creative solutions never proceed according to the inherently inductive path implied by the Thinking Process [8, 9]. Karl Popper and E. DeBono might agree that people like Goldratt, in reality, leap to a new hypothesis and then use the Thinking Process to justify it. Many people have followed through to develop the Thinking Process representations for TOC generic solutions, such as production and project management. The trees provide better and better understanding of those systems and may aid further improvement. They may also inhibit further improvement because they tend to put boundaries around the problem.

Understanding the thinking behind critical chain may help you discern whether your system conforms to the assumptions about current reality made in the process of developing critical chain. If your system differs in substantial ways, you may have to modify the future system and the injections to get a future reality that works for you.

The trees created as part of the Thinking Process are relatively complex and, thus, are not appropriate for publication in book format. You can find them on the Advanced Projects Internet site at http://www.advanced-projects.com.

11.3 Current Reality Tree

The CRT describes the system, as it is today, to help find the core conflict. The core conflict is a root cause of many UDEs. You start the CRT process with UDEs, which are those things that you feel "really bother me" about the current reality. For

example, "It really bothers me that projects overrun the schedule." Chapter 3 described the UDEs for project management.

You then select three of the UDEs to develop the core conflict. You do that by developing each UDE conflict and then combining the three conflicts to discern the underlying generic conflict that leads to all three. Chapter 3 also illustrates the combination of Evaporating Clouds for project management. The CRT derives the cause to a core conflict that leads to most (and usually all) of the UDEs. A core conflict is not the core conflict. It is an important conflict and, therefore, a good, high-leverage place to focus on changing the process.

The Evaporating Conflict Cloud: A Daily Tool!

In my opinion, the Evaporating Cloud is one of Goldratt's best inventions. It is a very powerful conflict resolution and decision making tool. You can use it to make decisions on your own. It helps you to understand the reasons why you are conflicted on a decision, and it helps you identify assumptions that will lead to a successful decision. You can use it to work our conflicts between yourself and others. You can use it to lead team-conflict resolution. You can use it to present your thinking on how to resolve conflicts. Hardly a day goes by in which I do not at least mentally think through the steps of the Evaporating Cloud for some problem or another.

Figure 11.3 illustrates the base of the project system CRT, containing the core conflict developed in Chapter 3 (see Figure 3.13). It illustrates the Evaporating Cloud in sufficiency tree format, highlighting the assumptions that lead to the conflict. Reading from the bottom, "If everyone wants projects to succeed, and if increasing competition drives managers and clients to demand projects to get the most scope for the least cost within the shortest schedule, then successful projects must deliver increased scope and reduced cost and schedule." Continuing up the tree, "If successful projects must deliver increased scope and reduced cost and schedule, and if the only way to reduce the schedule of critical-path plans is to reduce the duration of tasks on the critical path, and the only way to reduce cost is to reduce task cost, then there is pressure to reduce each task estimate."

You can read the right side of Figure 11.3 up to entity 9, which states, "There is pressure to include contingency in each task estimate. Including contingency conflicts with reducing task estimates, thus leading to the fights that surround project planning."

Note that I left out the phrase "more and more" in the above summary. The phrase becomes applicable as you traverse the tree over and over in the same organization. Note where some of the UDEs, which are conclusions farther up in the tree, feed into the entities at the base of the tree, such as UDE-1, -2, and -3 feeding all the way down to entity 9.

Figure 3.14 illustrates the notional connection of the UDEs that flow from the base of the tree. The logic of the tree is not evident at this summary level. The actual tree includes many steps of intervening logic that describe how organizational beliefs and actions lead from one UDE to the next. The key point of Figure 3.14 is that all the UDEs in the tree are causally related and derive from the core conflict.

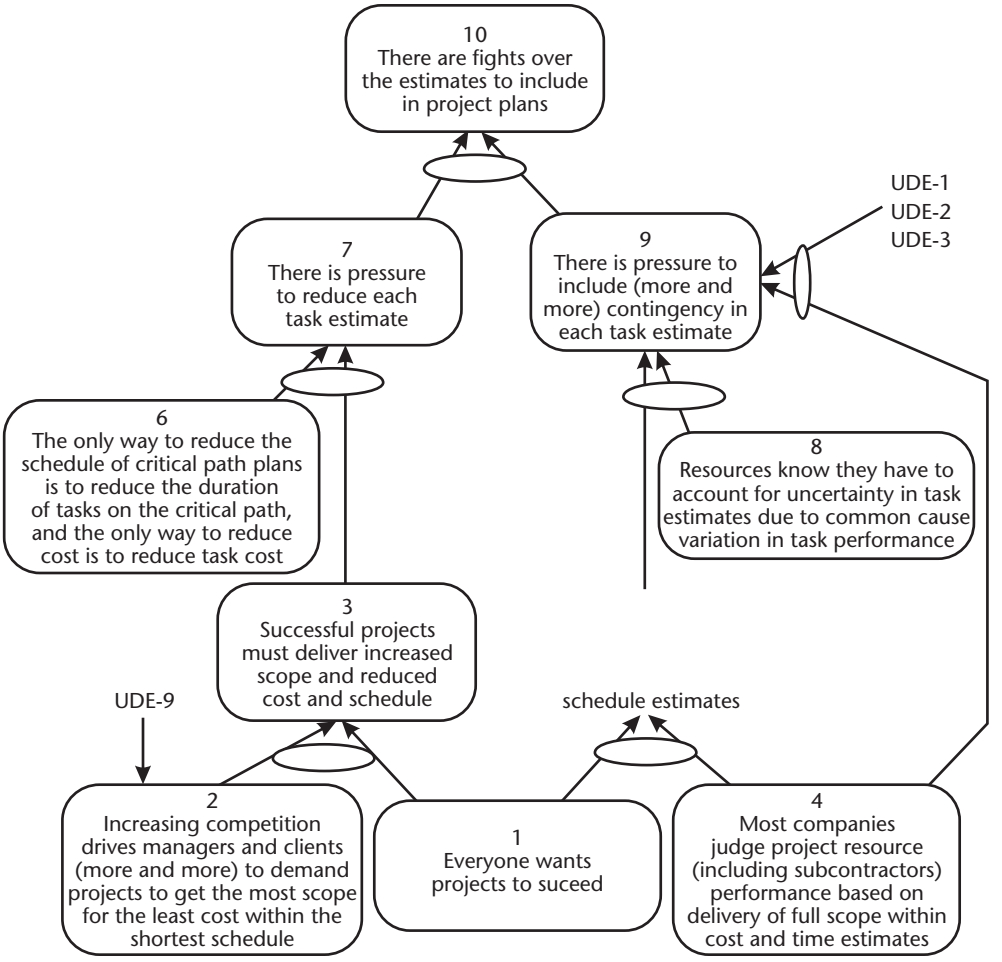


Figure 11.3 The single-project CRT chart base identifies the core conflict.

The generic project-management CRT cannot represent your environment. I have worked with companies that have started from very mature project-management systems, as well as with companies with a simplistic approach to project planning and control. Interestingly, it seems that organizations with the simplistic approaches are more able to adapt to CCPM. (Some argue that any degree of discipline would have helped as much. I dispute that claim because these are all highly multitasked environments, and conventional project management would only exacerbate that problem.) It is not unusual to find people whose idea of a project plan is a Gantt chart with no task logic and only a final due date demanded by someone outside the project organization. In a few months, it is possible to lead those people to have effective critical-chain project plans, buffer management, and a growing sense of success.

On the other hand, I have worked with organizations that have very detailed project plans (with thousands of activities) that, after six months, are still unable to resource-load their plans. In one case, the organization neglected to address the fact that the parts of the organization necessary to develop the complete project reported (collectively) to 13 different vice presidents in the organization, 12 of

whom had not bought in to the changes necessary to implement CCPM. Guess which vice president was replaced three months into the process. I think you can predict the result.

The CRT: Your Universal Problem Solver

You can apply the CRT to any problem you wish to solve. It is a great tool to gain stakeholder alignment on the problem to be solved on your project. You can also use it if your project gets into trouble to help uncover the core conflict that drives the current UDEs and to craft an effective recovery plan.

11.3.1 Policies, Measures, and Behavior

It is often useful to explicitly consider the impacts that policies and measures have on behavior and how that behavior impacts current reality. Policies and measures can often be the constraint of systems like project management. For example, entity 4 of the CRT, "Many companies judge project-resource performance based on delivery of full scope within cost and time estimates," may be reflected by performance-appraisal policies and measures. Very often, companies have efficiency measures and policies that reinforce multitasking. We have not explicitly included these on the CRT as they are very specific to each company and, therefore, difficult to put into a generic CRT. You should consider your company's policies that may reinforce the generic CRT.

Management Tip: Always Consider Policies, Measures, and Behaviors

Considering the impact of policies, measures, and behaviors is valuable for just about any management action you might think of. They become very important to create the feedback loops that cause and sustain change to new methods of doing business.

11.3.2 Feedback Loops

The detailed CRT logic contains feedback loops. For example, Figure 11.3 illustrated UDEs 1, 2, and 3 feeding into entity 9 at the base of the CRT. That circular nature of real systems bothers some people, who think in terms of cause effect reasoning and want to know, which comes first, the chicken or the egg? In the real world of dynamic systems, it is an ongoing circular pattern that evolves over time. That is one reason that people looking for a root cause often get the wrong answer; the cause is the overall structure of the system, not an individual entity. All variables in the system change in a correlated way (although often with time delays).

The feedback loops are often where you can find the most leverage to change the system and, thus, are worth considering explicitly. Your measurement systems almost always constitute a feedback loop (if they do not, why are you making the measurement?). The primary feedback key to CCPM is management's reaction to task performance relative to estimated 50–50 durations. Management's response to buffer reporting is also critical.

11.3.3 Scrutiny

Scrutiny critically reviews the products of the Thinking Process. It is the primary way to determine if the hypothesis of the Thinking Process connects with reality better than alternative processes. (The secondary method is the tried-and-true method of scientific experimentation, which is very difficult to perform with systems as complex as projects.) Chapter 3 describes how, using the scientific method as the TOC theory of knowledge, you can never prove the truth of any assertion or hypothesis. The best you can hope to do is, through critical review, assure yourself that one hypothesis describes reality better than another does.

Scrutiny provides the Thinking Process critical review. You do it by subjecting each product of the Thinking Process to the Categories of Legitimate Reservation. I learned these categories from the partners and associates at AGI, but I suspect that they have some prior source in logic. Unfortunately, Goldratt does not use references in his books, and AGI does not reference the source of much of their material. W. Dettmer [10] and others have published these categories but only refer back to AGI as the source.

The categories of legitimate reservation are:

- Clarity: Does everyone understand the meaning of the words in the entity?
- Entity existence: Is there evidence to support or refute the existence of the entity in reality?
- Causality existence: Is there evidence to support the claim of causality? (This evidence is usually of the form that the effect always exists when the cause is present and never exists when the cause is not present.)
- Cause insufficiency: Do other entities also have to be present to cause the stated effect?
- Additional cause: Can the effect exist without the stated causes or in the presence of other causes? (The "good-enough" TOC principle suggests you limit your model to causes that represent at least one third of the effect instances.)
- House on fire (or cause effect reversal): "If there is smoke, then there is fire" is a common expression that does not reflect causality; even though expressed as IF/THEN. The accurate statement is, "If I see smoke, then I know that the house is on fire." However, the fact of there being smoke does not cause the house to be on fire.
- Predicted effect: Would the existence of the entity also cause some other effect, and does that other effect exist in reality?
- Tautology: Ayn Rand's famous "A is A," or as the story goes,

> Joe: "blowing a horn every day keeps elephants out of my living room."
> Dan: "how do you know this?"
> Joe: "do you see any elephants in my living room?"

Detecting this logic error is sometimes hard. How many people call the psychics' 900 numbers?

Jonahs subject each Thinking Process tree to critical review for compliance with these criteria. If you think getting people to think through the trees is difficult, you

have to try going through the review process of each entity and causality in the tree. Most people find it agonizing. Unfortunately, there does not appear to be a better alternative.

11.3.4 Buy-in

The people who will deploy the results of the Thinking Process have to agree with it to the extent that they are willing to make or tolerate (depending on their location and effect on the system) the changes it requires to move to future reality. Unfortunately, experience demonstrates that logic rarely influences people's beliefs. As Section 9.2.5 describes, if you do not change beliefs, you are unlikely to succeed in changing behavior for the long term. You may cause a temporary impact, but the system will, over time, swing back to where it was before you started.

Buy-in seeks to achieve sufficient agreement to make the injections necessary for future reality. You will not achieve complete changes in belief at the outset, but if the future reality contains sufficient positive feedback to meet the real needs of the people who influence the system, this feedback will take over driving the system. You can achieve this with some people by leading them through the Thinking Process; but experience demonstrates that it is the rare senior manager who will focus long enough to follow this through. Thus, you will need to use specially designed tools, presentations, and dialog to reach these people.

I have found that most people intuitively appreciate the logic and use of the Evaporating Cloud. You can even get senior managers to listen to the three clouds that lead to CCPM, then to buy in to the core conflict and the necessary injections.

A powerful method to enhance buy-in is first to solicit the UDEs of those whose agreement you need and to include one or two of their UDEs in your CRT. Demonstrating how the system causes their UDEs usually succeeds in bringing about buy-in, not only to the CRT but also to the injections you propose to create future reality.

11.4 Future Reality Tree

The FRT describes "what to change to." It is your vision of the future. Future reality does not exist when you start to make the changes to eliminate the UDEs of current reality. The outputs of the FRT are the injections we have to create in order for future reality to exist. Injections are effects that, if in place, will cause future reality. Injections are not (generally) actions; you develop actions as part of how to cause the change.

11.4.1 Desired Effects

Start the FRT by changing each UDE into its opposite DE. Figure 11.4 illustrates the map of DEs for a successful project-management system, including an indication of where the injections tie in.

11.4.2 Injections

Injections are the changes you will make to the system. The FRT connects your injections to the DEs of future reality. You then methodically work through the tree,

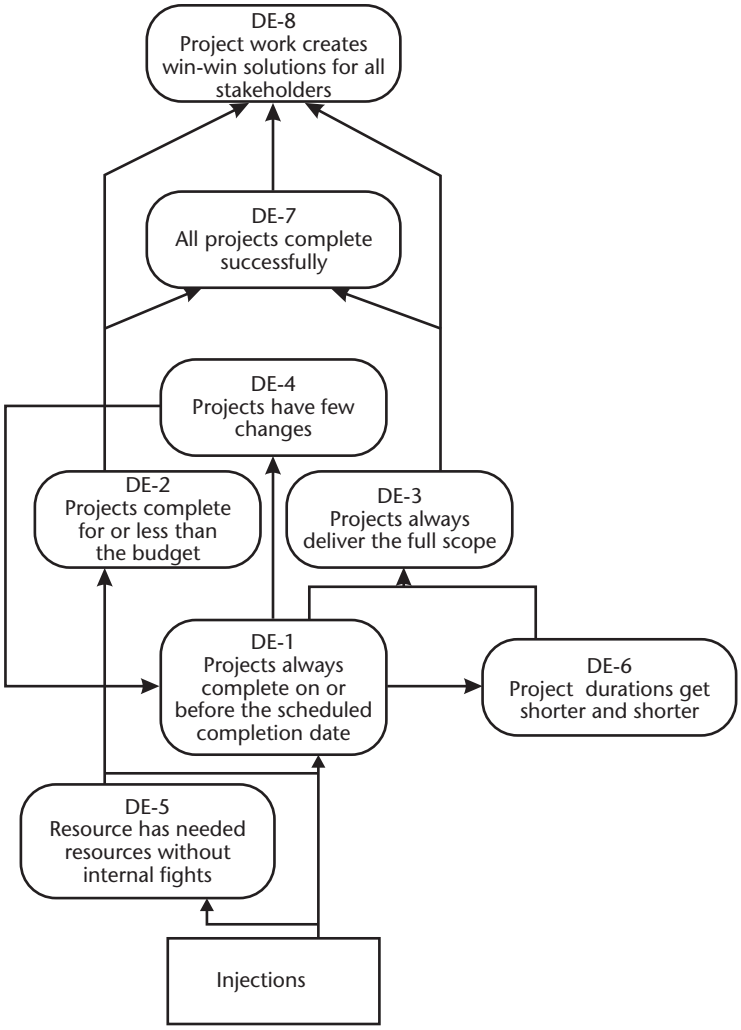


Figure 11.4 The DE map illustrates the relationship of DEs in the FRT.

determining where injections are necessary to create the future reality. Developing effective injections is the most creative stage of the Thinking Process. Experienced TOC experts like to word injections as completed effects, knowing that they will often have to develop a number of actions to achieve the injections. Collectively, the injections will create a future reality in which all the DEs exist.

The total list of injections follows:

- Reduce duration estimates by 50%. Project managers identify the project's network of activities and paths by unbuffered time and by resource. Activity durations are normal estimates, which we know to be high probability. We estimate the 50% probability duration by cutting these in half.
- Eliminate resource contentions and identify the critical chain. Project managers identify the critical chain as the longest chain of dependent events, including resolving resource contentions. This is the first focusing step.

- Insert a project buffer sized and placed to aggregate critical-chain contingency time (initially 50% of the critical-chain path length). This is one step to exploit the constraint.
- Ensure that resources work on the right task using prioritized tasklists or resource flags.
- Size and place feeding buffers on all paths that feed the critical chain. Project managers use the feeding buffers to immunize the critical chain from accumulation of negative variations on the feeding chains. This subordinates the other project paths to the constraint.
- The plan schedules activities to start as late as possible, protected by buffers. This injection helps to further subordinate the other paths to the constraint by allowing the critical chain (usually) to start first, with at most a few other paths.
- Resources deliver relay-racer performance (eliminate multitasking and the student syndrome). The resources work as quickly as possible, as soon as possible, on their activities and pass their work on as soon as they complete. No more student-syndrome work habits. This injection begins to elevate the constraint.
- The project manager provides resources with activity durations and estimated start times, not milestones. This injection helps to break the current win–lose paradigm associated with getting work done by the milestone date. It aids in encouraging resources to pass on their work when done. It aids in elevating the constraint.
- Project manager uses buffer management to control to plan. The project and feeding buffers provide the information to the project manager when to plan for recovery and when to take recovery actions. This is the first anticipatory measure ever used in project management! It also aids in elevating the constraint.

Figure 11.5 illustrates the injections in the notional overall sequence they appear in the FRT.

11.4.3 The FRT as a Guide for Change

The FRT becomes the guide for change. As you implement injections on the FRT, you use it to monitor whether you are achieving the DEs. You can also use the FRT as a resource to derive any unintended consequences of the proposed changes. Note that changes that occur more quickly or that are larger than predicted by your FRT are also a cause for reassessment since you may have missed some feature of the causalities.

The FRT provides a check on the CRT. You will often discover additional causalities that exist in current reality as you develop the FRT. Because the FRT focuses on the future, it is not necessary to go back and revise the CRT.

The bottom of the FRT starts with the injections summarized earlier. Each injection includes adjacent entities that describe why we need the injection and the logic explaining how the injection satisfies the need.

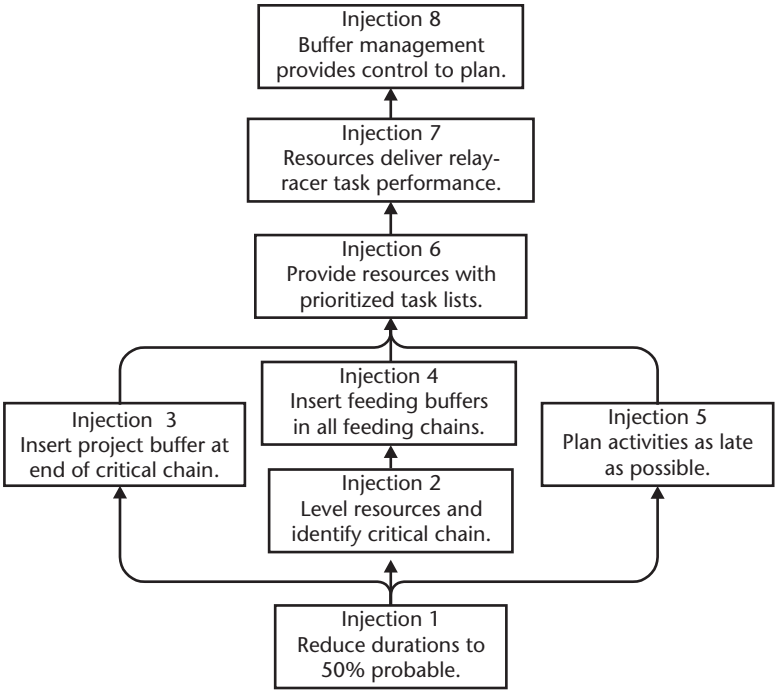


Figure 11.5 Injection map identifies the sequence of injections in the FRT.

11.4.4 Feedback Loops

The FRT contains the feedback loops to move the system to the new state and keep it stable. The project FRT exhibits a number of feedback loops, some of them with short delays and some of them with longer delays. The short-delay loops help to establish the system in the first place, and the long-delay feedback loops help to keep it stable.

One feedback loop illustrates the control effect of buffer management. When the project manager acts to restore a buffer to the yellow or green region, that will cause some activities to perform in a shorter time.

Another feedback loop shows that as teams build confidence by completing projects successfully with CCPM, they act to further reduce overall planned lead-time.

Experience indicates that the project feedback loop activates long before the first project completes, leading to further reductions in the initial project lead-time. That is likely due to increased confidence as the first part of the project shows little difficulty performing to the critical-chain schedule. The project manager contributes by not criticizing those who do overrun reduced-duration task times, as long as they exhibit relay racer performance.

11.4.5 Unintended Consequences (a.k.a. Negative Branches)

Unintended consequences are undesired results that occur from the actions we take. In a sense, they are an equal and opposite reaction to whatever action we have taken on the system. Sometimes (rarely, it seems), unintended consequences can be good. Often, however, they are not good. For that reason, we call the tool used to

understand and prevent or mitigate the potential negative effects a "negative branch," or NBR.

The distinction between obstacles and NBRs is that obstacles prevent you from achieving the future reality or ambitious goal you have set. NBRs come about because you have succeeded in creating the injection you intended to create. The injection, combined with other factors in current reality (or, sometimes, with new factors also created in future reality), conspire to cause a negative outcome.

The major resource for identifying potential NBRs is the people who review your FRT. They have the intuition to understand how the changes you are going to create may interact with their reality to create an unintended consequence.

Figure 11.6 illustrates a potential NBR dealing with the commonly voiced concern that if you make the safety time in the schedule evident, people will want to cut it. People usually include both customers and more-senior management. The project-management literature addresses that concern, often with a caution to keep your safety time hidden. That hardly seems like a professional way to run a business!

You first build the tree to connect the injections expected to cause the UDE to the stated UDE. The NBR is a sufficiency tree, just like the CRT and FRT, and is read IF/THEN. By this time, I trust you have sufficient comfort with the construction to read the tree. You must check the tree to ensure that the entities and causalities exist and that the logic is complete and sufficient.

The next step in using the NBR is to find an injection that will prevent the negative effect. You do that by examining the branch and locating the point at which it turns negative. In the case of this figure, the branch turns negative at entity 602, "The customer wants lead-time reduced by cutting buffer times." You then assess potential assumptions under the causality arrows that feed that entity. In this case, since only one causality feeds entity 602, you have to look at only two arrows. Note that since the connections from 600 and 601 to 602 include an "and" illustrated on the tree by flattened ellipse called a "banana" by TOC practitioner eliminating either 600 or 601 results in eliminating the effect, 602. That is the meaning of the "and." You need both feeding entities for the effect to follow.

In this instance, entity 601 appears to be a fact of life, so there would be little advantage to questioning the assumptions surrounding its existence or causality. There are assumptions in the causality between 600 and 602, the most obvious one being, as noted on the tree, "Because the customer sees cutting buffers as the biggest and most opportune way to meet their need." That is likely to be a true assumption when the customer (which may be internal management) does not understand the ideas behind CCPM.

Once you have an assumption, you can propose alternative injections that make the assumption no longer correct or applicable. Two potential injections are presented in Figure 11.6. Either injection should do the job of eliminating the assumption and therefore preventing the UDE of this NBR. Take your choice.

The NBR procedure follows:

1. Identify the potential UDE of concern.
2. Identify the injection you suspect leads to the UDE.

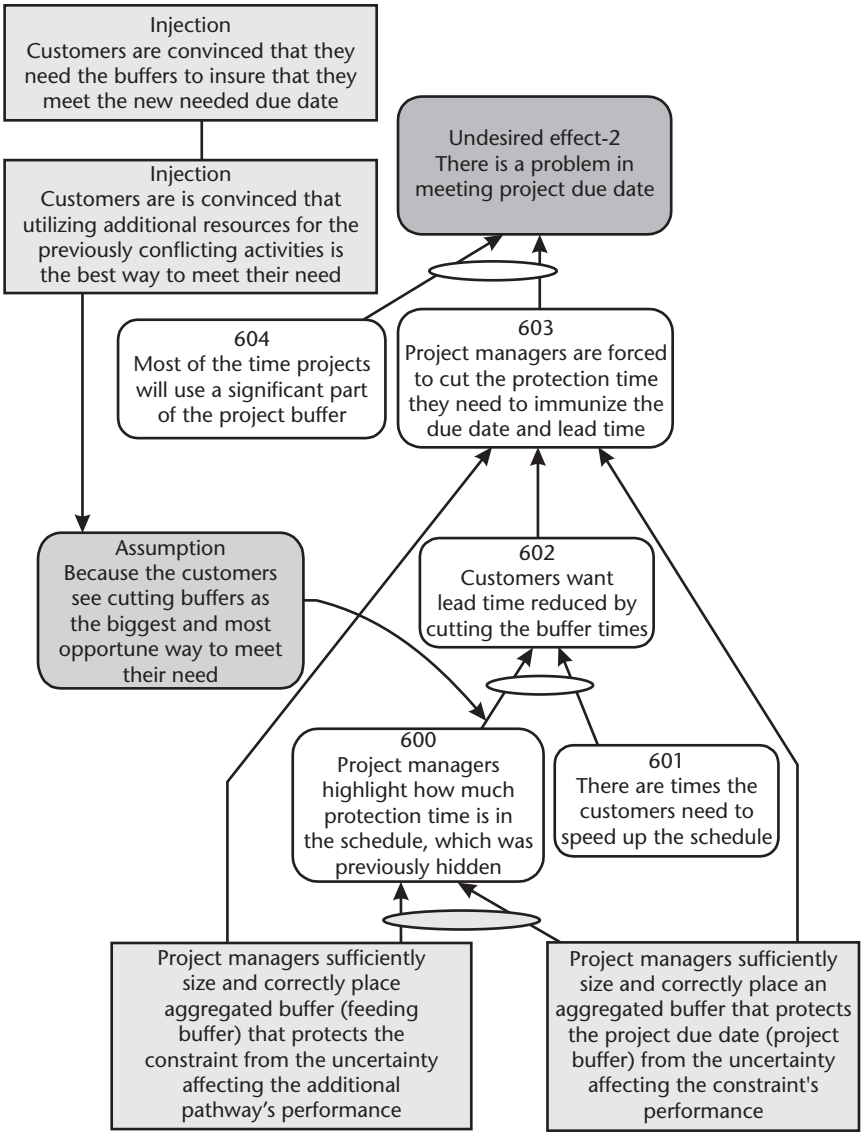


Figure 11.6 An NBR example.

3. Build a sufficiency tree to connect logically the injection to the UDE.
4. Scrutinize the logic in the tree (branch) by reading it aloud to others and having them agree to the logic.
5. Determine where the branch first turns negative.
6. Expose the assumptions under the arrows feeding the first negative entity on the branch.
7. Identify injections that will invalidate the assumptions and, therefore, prevent the negative effect.

It is, of course, possible that the injections you propose to trim the NBR may themselves lead to unintended consequences. If so, examine the new NBRs before completing the strategy.

Negative Branches for Risk Management

The NBR can be of great help in project-risk management. It helps a team clarify the causes of potential risk events and, thus, to identify prevention and mitigation measures to avoid or reduce the probability of those causes or to plan for mitigation should the risk event occur.

An injection is not a plan. It is not even a coherent strategy. Goldratt suggests the following tools for such purposes.

11.5 Prerequisite Tree

The PRT is a time-phased tree of the effects that we must cause for the FRT to result. You assess each injection on the FRT to determine the obstacles that must be overcome for the injection to exist. You create intermediate objectives to overcome the obstacles and logically link them in a time sequence with the injections. Figure 11.7 illustrates the PRT for the first critical-chain injection. The obstacles are shown in the hexagons.

Add each new injection to the overall PRT to accomplish future reality. Each injection must have a transition tree TRT to implement the actions necessary to achieve the injections and/or intermediate objectives that build to the injections.

The PRT also has stand-alone utility as a tool to plan and achieve ambitious goals. Goldratt illustrates its use for this purpose in *It's Not Luck* [11]. It has great power to get a team to identify all of the obstacles they foresee at the beginning of a project and to create a plan based on overcoming all of these obstacles.

PRTs for Task Networks and Work Breakdown Structures

Many have learned to use the PRT approach as a backward-planning approach to develop the project-task network. Instead of soliciting obstacles, start with deliverables and ask, what inputs do I need to create this deliverable? If you have to, insert a task to create that input and ask the question again. Keep asking the question until you have all the inputs you need for the project result. You can then ask for the obstacles to getting there and add resolving those obstacles to your scope of work. The higher levels of the PRT developed this way can make up your WBS. Just be sure the WBS specifies deliverables (results). The tasks (activities) in your network should have both a noun and a verb in the label, as they convert inputs to outputs through some action.

11.6 Transition Tree

The TRT provides the time-phased action plan to achieve the effects on the PRT. The TRT ties the actions to the logic for doing them and provides clear instructions to those who perform the activities. You can also use it to measure progress in terms of the effects produced, not the performance of the action. The TRT has broad application for achieving any effect you wish. (For example, you can use TRTs to get buy-in to the Thinking Process results.).

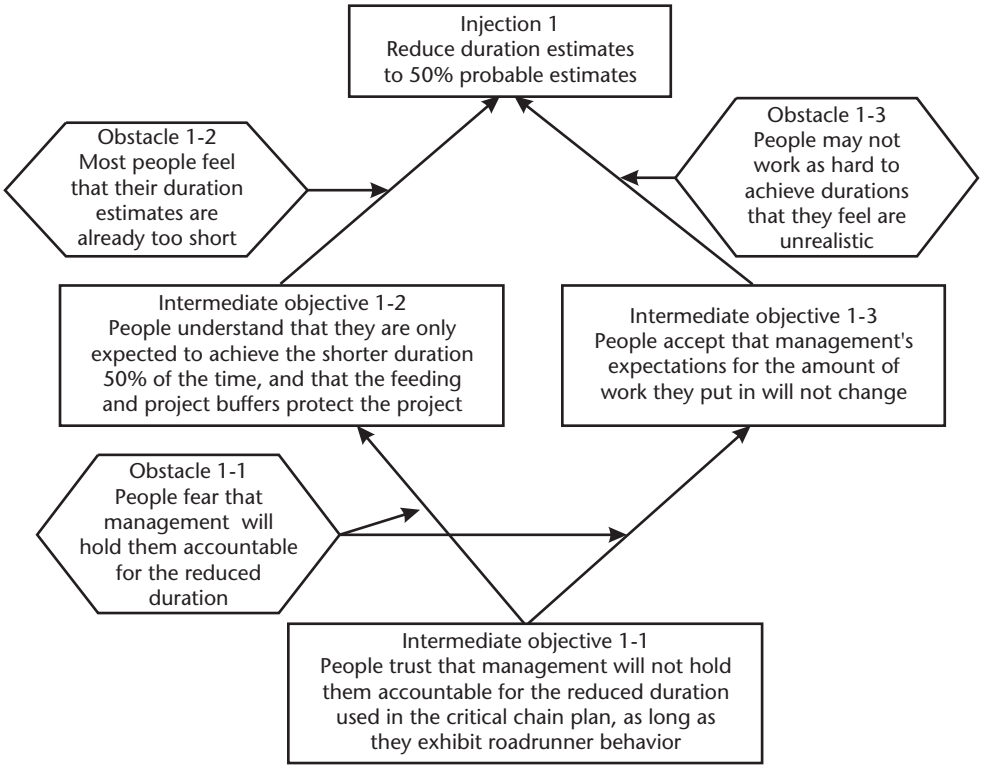


Figure 11.7 PRT example for the first injection.

Figure 11.8 illustrates a TRT for one of the project-management system injections. This tree creates the first intermediate objective in Figure 11.7. You should only create the TRT for two levels of the PRT at a time, starting from the bottom. As you complete the intermediate objectives at one level, you should create the TRT for the next higher level.

You read the TRT from the bottom up the same way as you read the CRT and the FRT. It is an IF/THEN tree. The TRT describes the logic for each action and why we expect the action to create the DE. It has proven a very effective tool for communicating clear instructions.

The TRT also has stand-alone utility as a way to present procedures.

Transition Trees Create Procedures That Work

The TRT seems like a very simple construct, but like many TOC tools, it has deceptive power. When people apply the TRT to procedures that they think they know, including written-down, formal procedures, they usually find mistakes in the existing procedures. The TRT requires that you specify the logic for the order of the actions, the expected result of the action, and why the action should cause that result. Most procedures do not include this detail.

Recently some suggestions have been offered to change the structure of the TRT linkages, but the suggested changes keep the overall logic and content as specified here. If a different format works for you, that is fine: it is the content that counts!

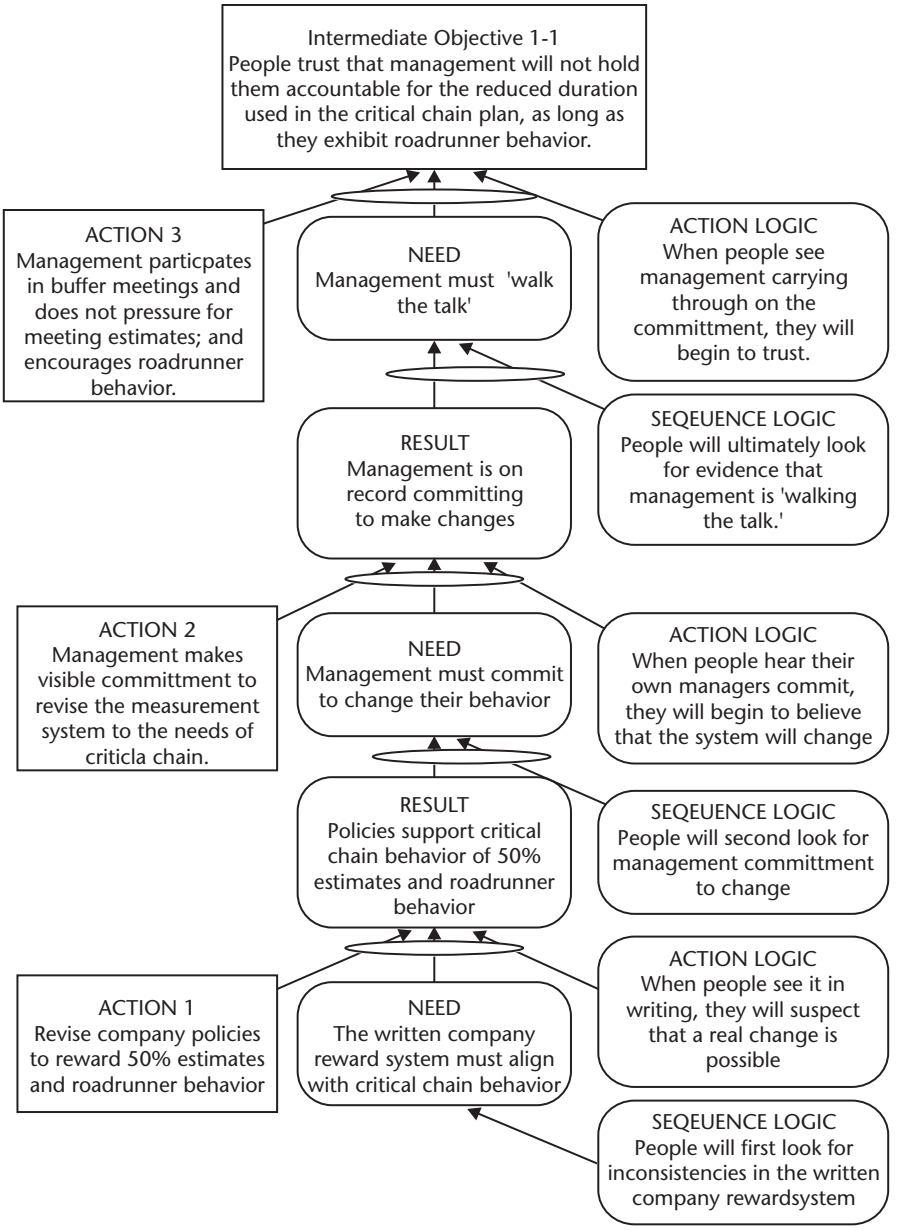


Figure 11.8 The TRT identifies the actions, effects, and logic to achieve the intermediate objectives. It provides clear instructions.

11.7 The Multiproject Process

11.7.1 Multiproject CRT Additions

The multiproject environment adds the following additional UDEs:

- Management commits to project dates that are unachievable.
- Project managers fight over resources.
- People have to work on multiple tasks for multiple projects.

- People are pressured by project managers to complete that project manager's task first.
- People are forced to work extensive, uncompensated overtime.

Figure 11.9 illustrates the CRT additions required for the social issues of multiprojects.

11.7.2 Multiproject FRT Additions

The multiproject FRT requires adding the following injections:

1. Identify the drum resource for the organization.
2. Identify the priority of projects for scheduling of the drum resource.
3. Develop project start and completion dates from the individual project plans and the drum-resource schedule.
4. Insert a capacity-constraint buffer between uses of the drum resource on projects.
5. Insert a drum buffer upstream of the use of the drum resource in each project.

11.7.3 Multiproject PRT Additions

The multiproject PRT therefore requires adding the following intermediate objectives:

1. The drum resource is selected.
2. The drum manager is identified.
3. Initial project priorities are assigned.

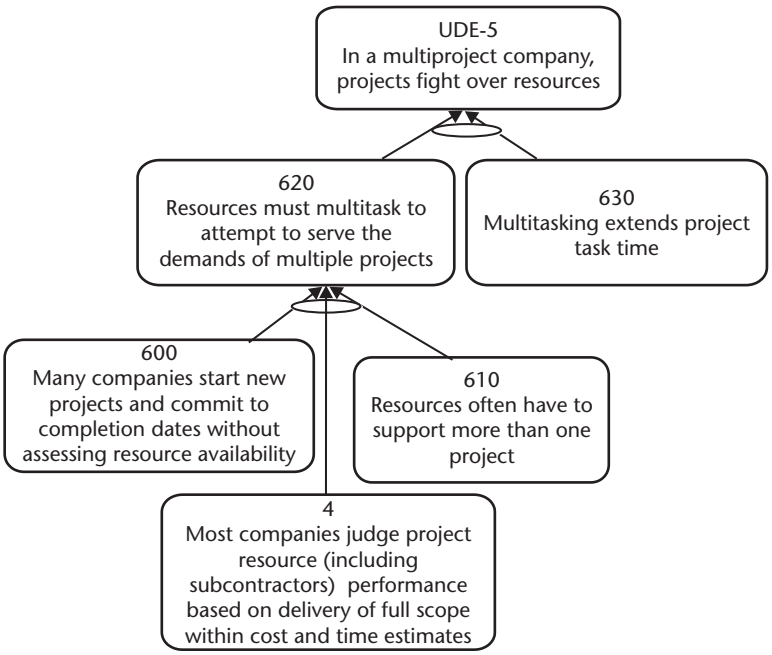


Figure 11.9 CRT additions for multiproject.

4. Management commits to new project-completion dates only after deciding the new project priority (relative to ongoing projects), developing the new project plan, and scheduling through the drum resource (pipelining).

11.8 Future Directions

Hopefully you have concluded by this point that the Thinking Process can apply to any problem. If you have, I agree with you! As such, you can bring it to bear on many problems of project management, beginning with deciding on a portfolio of projects to achieve some organizational goal. If you are a problem solver, you can begin with a CRT and the UDEs you wish to resolve. If you are a goal seeker, you can begin with the FRT and lay out a coherent strategy and synchronized plan to achieve your goals.

Chapter 2 presented theories that provide tools for ongoing improvement to your project-management process and to your entire enterprise. The TOC five focusing steps provide a strategy to continue improving a system. In addition, you can expand the definition of the system to widen the impact. The following are some areas that I see as fertile for future improvement:

- Expanding TOC thinking to the rest of the PMBOK™: This chapter has provided some ideas in that direction but more development awaits. The initial analysis to develop CCPM did not consider the PMBOK™. You can improve the remaining parts, including human resource management and project quality. Those areas are amenable to the Thinking Process to discover the core conflict and provide a plan for effective implementation.
- Extending TOC in areas it glosses over, such as human behavior, psychology, group dynamics, and creative thinking (e.g., adding in DeBono's ideas [8] and TRIZ (Russian acronym for Theory of Inventive Problem Solving)[12]): This might include extending TOC to other system archetypes described by P. Senge [13], beyond the limits to growth archetype, an unspoken assumption underlying the TOC.
- Improving user effectiveness on PMBOK™ processes: Many project teams struggle with portions of the project process, such as defining the WBS and creating the project network.
- Enhancing processes that address psychology and human behavior: TOC, the PMBOK™ and OPM3 are weak on these areas (although TOC provides some powerful tools, most notably the Evaporating Cloud for conflict resolution.) Six Sigma seems to have retreated from this aspect of Deming's system of profound knowledge.
- Continuing to build on applying Lean and Agile thinking to project management: CCPM has begun to exploit these ideas, but there is much more to be learned and applied. The Lean focus on eliminating waste seems especially valuable. The Agile approach to developing workable product at short intervals, while especially appropriate for software, may be extensible to other project types.

People pose new theories daily. Perhaps you have some? Some of these theories will extend their predecessors in the way Six Sigma has taken over where TQM left off. Others will take new approaches.

I personally feel that the greatest business and organizational improvements lie in improving management processes. I see a continuing trend toward more and more ineffective management as people inappropriately apply technological tools, such as pagers, cell phones, and e-mail. More and more interruptions mean less and less focus, leading to a frenzied pace and little accomplishment. I imagine some management as a fibrillating heart thrashing away while producing less and less output. If managers cannot manage themselves, how can they expect to effectively lead others? Most of the workers in their groups report increased multitasking, longer and longer work weeks, and more and more pressure.

I invite you to join me in continuing the journey. I continue to look for system improvements that step beyond the material presented in this book. These directions include better scheduling and decision-making support tools, improved methods to reduce the variation in project-task performance, and improvements that address the causes of frustration experienced by many project participants.

11.9 Summary

The Thinking Process helps you improve a system. As illustrated in the preceding chapters, the individual Thinking Process tools are amenable to individual application for specific purposes. You can use the Evaporating Cloud for a wide range of problems, from individual internal decision making to resolution of international issues. The following are key points from the TOC Thinking Process application to project management:

- The core conflict, leading to all project UDEs, is between the individuals and the project-management system.
- The core conflict derives from how the project system manages (or fails to manage) uncertainty by allocating contingency to buffers.
- The constraint for single projects is the critical chain, the longest path through the project considering both the project-task logic and the resource constraint.
- A system to exploit this constraint aggregates individual task uncertainty into buffers at the end of activity chains.
- Buffer management provides a real-time information (the answer to the question asked) system to effectively manage projects to complete at or before the scheduled end of the project buffer.
- The system constraint in a multiproject environment is a resource shared across multiple projects (the drum resource).
- Exploiting the multiproject resource constraint requires eliminating bad multitasking of all project resources.
- You must subordinate resource efficiency measures to the multiproject constraint. (Note that all resources, not just the drum, require protective capacity.)

You should modify the model as necessary to match the specific requirements of your environment. So far, the generic injections have proved to be robust over a wide range of project environments. Specific deployment (e.g., implementation plan, PRT, TRT) varies. Experience to date demonstrates that effective leadership and a good implementation plan are the critical success factors. Do not underestimate the cultural resistance that *will* occur. Plan to counter the resistance, but do not inflict harm on the resistors. Frequently, the greatest resistors become the biggest champions once they have understood and tried the new approach.

11.10 Conclusion

I ask you not to blindly accept a single thought I have presented in this book. Rather, consider the ideas and think them through for yourself and your organization. When you do so, I encourage you to think critically and identify as many potential obstacles and unintended consequences as you can. But, don't stop there. The *only* reason to identify obstacles and unintended consequences is to overcome and avoid them. I sincerely hope that the methods in this book will help you do so.

Thank you for joining me on this journey. My greatest hope is that you can improve the happiness of everyone on your project teams.

References

[1] Deming, W. E., *Out of the Crisis,* Cambridge, MA: MIT Press, 1982.

[2] Womack, J. P., and D. T. Jones, *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, New York: Simon and Schuster, 1996.

[3] Pande, P. S., R. P. Neuman, and R. R. Cavanagh, *The Six Sigma Way: Team Fieldbook*. New York: McGraw-Hill, 2002.

[4] PMI, *OPM3, Organizational Project Management Maturity Model*, Newton Square, PA: PMI, 2003.

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

[6] Noreen, Eric, Debra Smith, and James T. Mackey, *The Theory of Constraints and Its Implications for Management Accounting*, Great Barrington, MA: The North River Press, 1995.

[7] Goldratt, Eliyahu M., *The Goal*, Croton-on Hudson, NY: North River Press, 1984.

[8] DeBono, E., *Lateral Thinking, Creativity Step by Step*, New York: Harper & Row, 1970.

[9] Popper, K., *Objective Knowledge, An Evolutionary Approach*, Oxford: Oxford University Press, 1972.

[10] Dettmer, H. William, *Eliyahu M. Goldratt's The Theory of Constraints, A Systems Approach to Continuous Improvement*, Milwaukee, WI, University Bookstore, 1995 (now Available through ASQC Press).

[11] Goldratt, Eliyahu M., *It's Not Luck*, Great Barrington, MA: North River Press, 1994.

[12] Altshuller, G., *And Suddenly the Inventor Appeared: TRIZ, the Theory of Inventive Problem Solving*, Worcester, MA: Technical Innovation Center, Inc., 1996.

[13] Senge, P., et al., *The Dance of Change*, New York: Currency Doubleday, 1999.

Glossary

Activity The lowest level of the WBS. A packet of work that forms the basic building block of a plan or network.

Activity Network A network made up of two or more activities with dependency.

Actual Cost The actual money spent in performing an activity so far.

Actual Cost of Work Performed (ACWP) The CSCS term for actual cost.

Additional Cause Reservation A reservation applied to a Thinking Process tree suggesting that there may be an additional cause that create an effect at least one third of the time.

Banana Title for the logical "and" on a Thinking Process tree, represented by a flat oval connecting the causality arrows to be "anded."

Bar Chart See Gantt Chart. A chart of the known watering holes of the project team. For use when a crisis occurs after normal working hours.

Body of Knowledge A document produced by the PMI, which defines areas of knowledge that make up the discipline of project management.

Bottleneck The constraint in a production flow process. The capacity-limiting process step. The critical chain is a single-project bottleneck. A company may also have company resource constraints.

Budget The planned cost for a project, or the BCWS. It may also include a cost buffer, or a budget contingency.

Budgeted Cost of Work Performed (BCWP) Earned value of work done, equal to the amount that was budgeted for the activities completed.

Budgeted Cost of Work Scheduled (BCWS) The value of work that should have been completed by the current date according to the baseline plan.

Buffer Time or budget allowance used to protect scheduled throughput, delivery dates, or cost estimates on a production process or project. Buffers are sized based on the uncertainty in the protected group of activities. Therefore, the schedule buffers are not the same as float, or slack, which occurs as an accident of the activity logic in critical-path schedules.

Buffer Penetration The amount of the buffer that has been used up by actual progress in the project.

C/SCSC See Cost-Schedule-Control-Systems Criteria.

Capacity Constraint Resource In a multiproject environment, the resource that is most often overloaded.

Capacity Constraint Buffer Buffer placed between projects or in the drum schedule to sequence projects in a multiproject environment.

Categories of Legitimate Reservation (CLR) A set of logical tests for trees created by the Thinking Process.

Causality Reservation Questioning if an effect at the head of a causality arrow in a Thinking Process tree is an unavoidable consequence of the entity or entities at the tail of the arrow.

Cause An entity that inevitably leads to a certain result (effect). Causality is determined if the predicted effect is always present when the cause is present and never present when it is not. Causes may be single or may require other conditions to lead to the effect.

Clarity Reservation One of the legitimate reservations for scrutinizing Thinking Process trees. It means the reviewer does not fully understand the entity.

Cloud (Evaporating) A fixed-format necessity tree used to develop win–win solutions to action alternatives or conflicting wants. The action alternatives are best expressed as opposites (e.g., "do D; don't do D"). The cloud has five entities and arrows (see the Thinking Process description). You identify the assumptions underlying the arrows to resolve the cloud. You develop injections that will invalidate the assumption and, therefore, invalidate the arrow and "dissolve" the cloud.

Conflict Resolution Diagram (CRD) An alternative title for the Evaporating Cloud.

Common Cause A single entity that causes several effects.

Common-Cause Variation: Variation of process output that is within the capability of the process and, therefore, not assignable to a special cause. Also called natural variation.

Constraint A process or process step that limits throughput.

Core Problem A primary cause of most of the undesired-effect symptoms in your system. You identify the core problem as an entry point on your CRT that traces, in cause effect cause relationships, through at least two thirds of the undesired effects, and which you have the stamina and energy to change.

CPM The Critical-Path Method, which was the original innovation in using networks and defining a critical path through the network.

Communication The effective transmission of information so that the recipient understands clearly what the receiver intends. Communication media may take several forms: oral, written, textural, numerical, graphic, body language, paper, electronic, physical, and so forth.

Communication Current Reality Tree (CCRT) A special form of the CRT developed to communicate the tree for buy-in. It displays the evaporating cloud for the system at the bottom and builds to the undesired effects.

Communication Future Reality Tree (CFRT) A special form of the FRT developed for buy-in of a specific group of people. It shows the connection between the injections and specific benefits for the group addressed.

Conflict Management The art of managing conflict effectively. In the Thinking Process, this involves the Evaporating Cloud, Communication TRTs, and, for chronic conflicts, the negative branch.

Constraints In a project, the generic term for factors affecting the possible start and finish dates of an activity, including logic and imposed dates. In TOC, it is the factor that limits the system from obtaining more throughput.

Constraint Buffer A buffer placed in the drum resource schedule to ensure that the company constraint has sufficient protective capacity. Also known as a capacity constraint buffer.

Contingency A buffer.

Core Conflict The conflict that leads to the core problem.

Core Problem A problem that causes at least two thirds of the undesired effects in a CRT and that you have the stamina and energy to reverse. The core problem is often the root cause of a number of root causes, or the common cause.

Corrective Action A process for correcting defects by identifying the defect, assigning responsibility, performing causal analysis, planning a resolution, and implementing the resolution.

Cost Buffer The financial contingency added to a project to protect the overall project cost. As with schedule buffers, it is best to accumulate all of the individual activity cost contingencies into one, which will be much smaller than the sum of the individual buffers.

Cost Variance (CV) The value of the work done less the actual cost of the work done (i.e., BCWP – ACWP). A negative number shows the project is currently over budget.

Cost World A business perspective that focuses on reducing cost as the path to business success. The Cost World is typified by a belief that cost savings are additive.

Cost-Schedule-Control-Systems Criteria In 1967 the U.S. DOD defined a standard for the use of earned-value analysis in defense projects. It has since been adopted much more widely and is supported by most planning software.

Critical Activity An activity on the critical chain.

Critical Chain The longest set of dependent activities, with explicit consideration of resource availability, to achieve a project goal. The critical chain is *not* the same thing you get from performing resource allocation on a critical-path schedule. The critical chain defines an alternate path, which completes the project earlier by resolving resource contention up front.

Critical-Chain Feeding Buffer (CCFB) A time buffer at the end of a project activity chain which feeds the critical chain.

Critical-Chain Resource Buffer (CCRB) See Resource Buffer.

Critical-Chain Schedule A late finish schedule controlled by the critical chain, including a critical-chain completion buffer, critical-chain feeding buffers, and critical resource buffers.

Critical-Chain Project Management (CCPM) A project-management system that addresses all of the undesired effects from the project-management Current Reality tree. It includes a critical-chain plan, the flush measurement tool, buffer management, and relay racer task performance.

Critical Path The longest sequence of activities in a network. Usually, but not always, a sequence with zero float. The critical path is an accident of arithmetic. It may be the longest sequence of activities, but there may be others that have such minimal

float as to be inconsequential. It also does not account for resource constraints. Note that once resource leveling has been performed, slack, and hence the critical path, are no longer valid calculations. The PMI's definition of critical path notes that it will change as the project progresses.

Cost Schedule Control System A system for evaluating the work completed on a project as a basis for progress payments. The primary innovation is the use of the budgeted cost of work performed, that is, the estimate for an activity, as the measure of work completed.

CSCS Cost Schedule Control System.

CS² Cost Schedule Control System.

Current Reality Tree (CRT) A logical representation of the current business system under analysis, demonstrating how the core conflict connects to the system's undesired effects.

CV See Cost Variance.

Data Any string of characters that describes something about our reality.

DBR Drum–Buffer–Rope method for production scheduling. The drum is the capacity of the plant constraint and is used to set the overall throughput schedule. The buffers are in process inventories strategically located to eliminate starving the constraint due to statistical fluctuations. The rope is the information connection between the constraint and material release into the process.

Dependency Links The various types of links connecting Activities in a precedence network. They include finish to start, start to start, finish to finish, and start to finish.

Dependent Events Events or effects that are related in magnitude, time, or some other factor, such that they influence each other or have a common-cause influence.

Desired Effect (DE) The positive effect you want to have in future reality to replace your undesired effect of current reality.

Dollar Days The integration of the product of daily cash flow (in minus out) times the number of days. See Flush.

Drum The resource selected for sequencing projects. In production, the bottleneck processing rate used to schedule an entire plant. See Capacity Constraint Resource.

Drum Buffer A buffer placed in a project plan to ensure that the company constraint is not starved. Also known as a constraint resource buffer, strategic resource buffer, or as a constraint buffer.

Duration The amount of elapsed time an activity is estimated to take.

Early-Finish Date The earliest date by which an activity can finish. Calculated during the forward pass of critical-path analysis.

Early-Start Date The earliest date by which an activity can start. Calculated during the forward pass of critical-path analysis.

Earned Value The value of the work done, where value is calculated in terms of the baseline cost. Known as budgeted cost of work performed in the cost-schedule-control-systems criteria.

Earned-Value Analysis The analysis of project progress where the actual money spent is compared to the value of the work achieved. See also Cost-Schedule-Control-Systems Criteria.

Effect An entity representing the result of one or more causes.

Efficiency A measure of the speed and effectiveness with which a resource delivers a particular skill or a measure of how much time resources charge to projects versus "unbillable time."

Elevate The TOC term for increasing the throughput capability of the system constraint. For projects, this usually means adding resources.

Entity A condition that exists.

Entry Point An entity on a sufficiency tree that has no causes (arrows) leading into it.

Erroneous Information A wrong answer to the question asked.

Estimate at Completion The current estimated total cost of the project.

Estimate to Complete (ETC) An estimate of the time and/or effort required to complete the activity.

Estimating The process of developing the planned cost and duration for activities.

ETC See Estimate to Complete.

Evaporating Cloud See Cloud.

Existence Reservation This means, "Prove it," and can be applied to an entity or causality arrow in a Thinking Process tree.

Exploit The TOC term for assuring that the system makes most effective use of a constraint in terms of the system goal.

External Constraint A constraint that acts upon activities within a network from outside the network, typically a regulation, imposed date, or environmental condition.

Feeding Buffer See Critical-Chain Feeding Buffer (CCFB).

Finish to Start A type of dependency link in precedence networks, which indicates that the start of the successor activity may not occur until the predecessor activity has finished.

Five Focusing Steps The five-step process to identify and elevate constraints. See Section 2.6.3.

Float A measure of the time flexibility available in the performance of an activity. Available in three flavors: total float, free float, and independent float .The minimum amount of time by which an activity will be extended due to factors outside the project manager's control. See Slack.

Free Float The amount of time an activity may be delayed without causing delay to successor activities.

FRT See Future Reality Tree.

Future Reality Tree (FRT) A sufficiency tree connecting injections to desired effects.

Gantt Chart A chart showing a list of activities represented by bars that are proportional in length to their duration. The bars are positioned along a horizontal time scale.

Goal, The See Jonah.

Hockey Stick The shape of a curve that is relatively flat and then rises rapidly, representing, for example, the amount of effort one puts out as a deadline approaches.

House-on-Fire Reservation Original definition based on the logic statement, "IF there is smoke and fire engines, THEN the house is on fire." The smoke and fire engines

are not the cause of the house being on fire, but rather cause us to know that the house is on fire. Since the Thinking Process trees are EffectCauseEffect trees, house-on-fire logic is not correct.

Identify The first step of the TOC focusing process, consisting of identifying the system constraint.

Information An answer to the question asked.

Injection An action or effect that will be created in the future to change system performance.

Integrated Plan A plan combining cost and schedule to complete a project.

Intermediate Objective (IO) An action or effect that is a necessary prerequisite to an injection or another IO.

Invalid Data Data that is not needed to deduce the specific desired information.

Inventory All of the investment in the equipment necessary to convert raw material into throughput.

Jonah A title bestowed upon those who complete the AGI Jonah course and are, therefore, prepared to go forth and replenish the rain forests with trees. A leading character in Goldratt's book *The Goal*, Jonah is a teacher and leader in the Socratic tradition.

Late-Finish Date The latest date by which an activity can finish. Calculated during the backward pass of critical-path analysis. All activities in a critical-chain schedule use this date, except those that are moved forward in time to resolve resource contention.

Leadership Doing the right things and getting others to follow.

Linked Projects A term used in some computer packages to indicate projects that use a common set of resources.

Logic Link See Dependency Links.

Logic Loop A circular sequence of dependency links between activities in a network.

Mean The average of a group of data, also called the first moment of the data. In a distribution skewed to the right, like most duration and cost estimates, the mean is higher than the median.

Median The middle value in a group of ordered data.

Merge Node A node in a network diagram where two or more links or activities merge.

Milestone An activity of zero duration that represents a significant deliverable or stage of the project.

Milestone Plan A plan containing only milestones highlighting key points in the project.

Multiproject Management The art and science of managing multiple projects that are in some way interconnected. This may be through logic connections or, more likely, the use of common resources.

Multitasking Performing more than one project activity at the "same" time.

Need The requirement(s) that *must* be met in order to achieve an objective or goal.

Necessity Tree A logic tree in which each item at the tail of an arrow *must* exist in order for the item at the head of the arrow to exist because of some assumption or obstacle represented by the arrow.

Necessary Condition #1 Satisfy customers now and in the future. (A necessary condition to meet the goal of any enterprise.)

Necessary Condition #2 Satisfy and motivate employees now and in the future. (A necessary condition to meet the goal of any enterprise.)

Negative Branch A sufficiency logic tree (potential FRT) stemming from an injection, which may lead to undesirable effects.

Network A diagram in which the logical relationships between activities is shown in either activity on arrow or precedence format.

Network Analysis A generic term for analyzing networks, including PERT and critical-path analysis.

Node The start and end of activities in an activity in arrow network or the activity box in a precedence network.

Obstacle An entity that prevents an effect from existing.

Operating Expense All of the money it costs to convert raw material into throughput.

Overtime Extra time available for a resource that may be used as part of resource scheduling in some computer packages.

Percentage Complete A number estimating the amount of an activity that is finished. One of the ways of allocating BCWP.

Performance Measurement This is the method used to relate physical progress achieved with cost status. The method identifies whether cost variances are due to differences in the value of the work being performed, or because the work is costing more or less than estimated. In this way, it is possible to determine if a project is ahead, on, or behind budget. See Earned-Value Analysis.

PERT See Program Evaluation and Review Technique.

Pessimistic Duration The longest of the three durations in the three-duration technique, or PERT.

Plan A generic term used for a statement of intentions, whether they relate to time, cost, or quality in their many forms.

PMI Project Management Institute.

PMP Project Management professional.

Predecessor An activity that logically precedes the current activity. See also Successor.

Predicted Effect Reservation One of the Categories of Legitimate Reservations. This means, "That entity can't be right because if it existed, we would see another predicted effect."

Prerequisite Tree (PRT) A logic tree representing the time phasing of actions to achieve a goal, connecting intermediate objectives with effects that overcome obstacles. The PRT is read, "In order to have *entity at head of arrow*, we must have *entity at tail* of arrow because of *obstacle*."

Priority A means of defining the order in which activities will be scheduled during resource scheduling.

Probability Usually used in the context of risk as a measure of the likelihood of a risk occurring.

Process A sequence of interconnected activities, each of which has an input and an output.

Problem A gap between what we want and what we have.

Program A portfolio of projects selected and planned in a coordinated way so as to achieve a set of defined objectives, giving effect to various (and often overlapping) initiatives and/or implementing a strategy. Alternatively, a single large or very complex project or a set of otherwise unrelated projects bounded by a business cycle.

Program Evaluation and Review Technique A network-scheduling tool, initially distinguished from CPM by allowing and using three activity-duration estimates.

Program Management The selection and coordinated planning of a portfolio of projects so as to achieve a set of defined business objectives and the efficient execution of these projects within a controlled environment, such that they realize maximum benefit for the resulting business operations.

Program Manager The individual responsible for day-to-day management of the program.

Program Plan A plan for a program of projects. Distinguished from a program-management plan in that a program plan need not supply the management systems.

Progress Reporting The process of gathering information on work done and revising estimates, updating the plan, and reporting the revised plan.

Project A temporary management environment that is created in order to achieve a particular business objective through the control and coordination of logistical and technical resources.

Project Buffer A time buffer placed at the end of the critical chain in a project schedule to protect the overall schedule.

Project Management The managerial task of accomplishing a project on time, in budget, and to technical specification. The project manager is the single point of responsibility for achieving this.

Project Manager The person appointed to take day-to-day responsibility for management of the project throughout all its stages.

Quality According to Dr. Joseph Juran, "Fitness for use." Defined in terms of both a lack of defects and product features. According to Phillip Crosby, "Conformance to customer requirements." According to W. Edwards Deming, "A product or service possesses quality if it helps somebody and enjoys a good and sustainable market."

Required Data The data needed by the decision procedure to derive information.

Resource Entity that performs project work, including a person, contractor, or machine.

Resource Buffer (or Flag) A flag placed on the critical chain to ensure that resources are available when needed to protect the critical-chain schedule. This flag is insurance of resource availability and does not add time to the critical chain. It takes the form of a

contract with the resources that ensures their availability, whether or not you are ready to use them then, through the latest time you might need the resource. (Often called a critical-chain resource buffer [CCRB]).

Resource Leveling The process of rescheduling activities such that the requirement for resources on the project does not exceed resource limits.

Resource Limit The amount of a particular resource available to the project at a given point.

Risk The combination of the probability and consequence of an undesired outcome. Project risk usually denotes undesired outcome relative to the project scope, cost, or schedule. Other risks sometimes important to projects include safety, environment, business, and security risks.

Root Cause The cause, which if changed, will prevent recurrence of an undesired effect.

Rope The information flow from the drum (bottleneck or constraint resource) to the front of the line (material release), which controls plant production.

Schedule A collection of reports showing the timing of activities.

Schedule Variance (SV) The value of the work done less the value of the work that should have been done (i.e., BCWP – BCWS). A negative number shows the project is behind schedule.

Scheduling Determination of the best means of achieving a project's general and specific schedule objectives. This involves identification and optimization of the project's overall schedule requirements, resource availability, internal and external constraints, and activity sequencing.

Scrutiny Inspection of a tree to ensure that none of the categories of legitimate reservation apply and that all of the entities are necessary to connect the undesired effects.

Slack Free time in a critical-path schedule resulting from paths shorter than the critical path. See Float.

Special-Cause Variation Variation in the output of a process that has an assignable cause.

Statistical Fluctuations Common-cause variations in output quantity or quality, including activity duration and cost.

Student Syndrome The natural tendency of many people to wait until a due date is near before applying full energy to complete an activity. Also see Hockey Stick.

Sufficiency Tree A tree construction in which the existence of the entities at the tail of the arrow make the existence of the entity at the head of the arrow an unavoidable result.

Subordinate The third step in the TOC five-step focusing process, placing considerations not related to the company goal at a lower level of importance than items that directly affect the system's ability to achieve the goal.

Successor An activity that logically succeeds the current activity. See also Predecessor.

SV See Schedule Variance.

System A network of interdependent components that work together to accomplish the aim of the system. Without an aim, there is no system.

Systems and Procedures Detail the standard methods, practices, and procedures for handling frequently occurring events within the project. These will cover management approvals, controls, and technical requirements. Systems will also cover methods of handling information transfer and storage.

Task A term usually synonymous with activity.

Theory of Constraints (TOC) A system theory developed by Dr. Eliyahu Goldratt and first published in his book *The Goal*. The most basic statement of the theory is that the output of a system is limited by a constraint.

Throughput All of the money our customers pay us minus the raw material cost.

Thinking Process The five-step process that identifies what to change, what to change to, and how to cause the change. The sequence of steps, starting with the CRT, through the Evaporating Cloud, FRT, PRT, and resulting in the TRT.

Transition Tree (TRT) An plan specifying the effects to be achieved, the starting conditions, the actions necessary to create the effects, the logic of why the actions will create the effects, and the logic for the sequence of the effects.

Variance (Statistical) A measure of the dispersion of a sample and estimate of the standard deviation of a population.

Want The effect that one believes *must* exist in order to satisfy a need because of some set of assumptions.

WBS See Work Breakdown Structure.

Work Breakdown Structure (WBS) A tree diagram that breaks a project down into increasing levels of detail. The lowest level of the WBS comprises work packages.

List of Acronyms

ABC	antecedent behavior consequence
ACWP	actual cost of the work performed
AGI	Avraham Goldratt Institute
ASAP	as soon as possible
BAC	budget at completion
BCWP	budgeted cost of work performed
BCWS	budgeted cost of work scheduled
CCPM	Critical-Chain Project Management
CMM™	Capability Maturity Model™
CPI	cost-performance index
CPM	Critical-Path Method
CRT	Current Reality Tree
CSCS, or CS^2	cost-schedule-control systems
CV	cost variance
DBR	Drum–Buffer–Rope
DE	desired effect
DMAIC	define measure analyze improve control
DOD	Department of Defense
DOE	Department of Energy
EAC	estimate at completion
F&OR	functional and operational requirements
FRT	Future Reality Tree
NBR	negative branch
OPM3	Organizational Project Management Maturity Model
PDCA	plan do check act
PERT	Program Evaluation and Review Technique
PMBOK™ Guide	*Guide to the Project Management Body of Knowledge*
PMBOK™	Project Management Body of Knowledge
PMI	Project Management Institute

PMPs	project-management professionals
PRT	Prerequisite Tree
R&D	research and development
RDU	remaining duration
RUP	Rational Unified Process
SEI	Software Engineering Institute
SIPOC	supplier input process output customer
SOWs	statements of work
SSQ	square root of the sum of the squares
SV	schedule variance
TOC	Theory of Constraints
TQM	Total Quality Management
TRT	Transition Tree
UDEs	undesired effects
WBS	work breakdown structure

About the Author

Lawrence P. Leach is the president of the Advanced Projects, Inc. (API), a management consulting firm. API specializes in project management, including leading the implementation of the new critical chain method of project management. Mr. Leach supports many large and small companies with diverse projects, including new product R&D, pharmaceuticals, IT, repair/maintenance, and construction. He worked at the vice president level in several Fortune 500 companies, where he managed programs of various large and small projects. Prior to that, he successfully managed dozens of projects, ranging up to one billion dollars.

Mr. Leach has masters' degrees in both business management from the University of Idaho and mechanical engineering from the University of Connecticut. He was awarded membership in Tau Beta Pi, the engineering honorary society, while earning his undergraduate degree in Engineering at Stevens Institute of Technology. Mr. Leach is a member of the Project Management Institute (PMI) and a certified project management professional (PMP). His has published many papers on related topics, including a *PMI Journal* article on critical chain in June 1999, a pair of papers published in *PM Network* in spring 2001, and a new *PM Journal* article in the June 2003 issue. He presents seminars for PMI Seminars and serves on the part-time faculty for the University of Phoenix, facilitating M.B.A. courses.

Index

Q

T

Recent Titles in the Artech House Effective Project Management Library

Robert K. Wysocki, Series Editor

Critical Chain Project Management, Second Edition, Lawrence P. Leach

The Project Management Communications Toolkit, Carl Pritchard

Project Management Process Improvement, Robert K. Wysocki

For further information on these and other Artech House titles,
including previously considered out-of-print books now available through our
In-Print-Forever® (IPF®) program, contact:

Artech House
685 Canton Street
Norwood, MA 02062
Phone: 781-769-9750
Fax: 781-769-6334
e-mail: artech@artechhouse.com

Artech House
46 Gillingham Street
London SW1V 1AH UK
Phone: +44 (0)20 7596-8750
Fax: +44 (0)20 7630-0166
e-mail: artech-uk@artechhouse.com

Find us on the World Wide Web at:
www.artechhouse.com