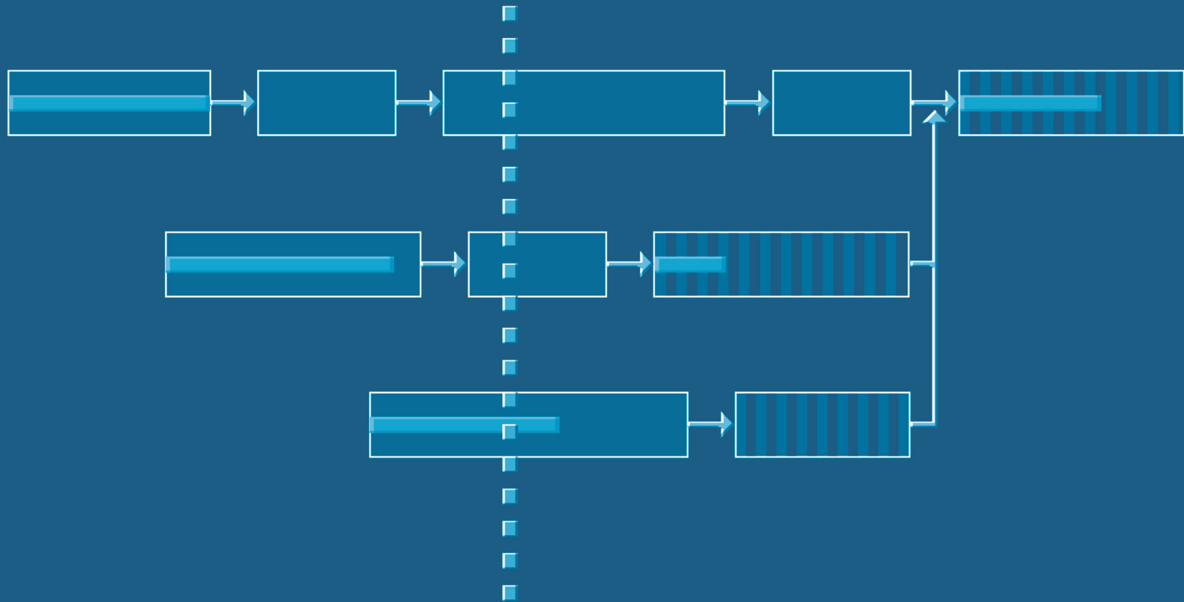


CRITICAL CHAIN PROJECT MANAGEMENT

THIRD EDITION

LAWRENCE P. LEACH

Critical Chain Project Management

Third Edition

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

Critical Chain Project Management

Third Edition

Lawrence P. Leach

ARTECH HOUSE
BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data
A catalog record for this book is available from the British Library.

ISBN-13: 978-1-60807-734-2

Cover design by Vicki Kane

10 9 8 7 6 5 4 3 2 1

Contents

Preface

Critical Chain Project Management (CCPM) has come a long way since the first edition of this book (published in 2000) and from when Dr. Eliyahu Goldratt published his book *Critical Chain* in 1997. William James wrote, "First, you know, a new theory is attacked as absurd; then it is admitted to be true, but obvious and insignificant; finally it is seen to be so important that its adversaries claim that they themselves discovered it." CCPM was in James's first phase when the first edition of this book was published in 2000.

Critical chain scheduling has since moved well beyond James's first phase. It has now appeared in three editions of the Project Management Institute's *Guide to the Project Management Body of Knowledge* (PMBOK™ Guide), is treated in their *Practice Guide to Scheduling* and *Practice Guide to Risk Management*, and is now a topic in most project management texts. There have been thousands of successful CCPM projects in different national cultures, business cultures, industry groups, product lines, and company sizes. Many organizations have standardized their processes to include it and have made great gains in project delivery as a result. So one might say that it passed James's second phase because many say it is obvious and its presence in so many places shows its significance.

However, CCPM has yet to become the standard in industry. In some respects it still seems to qualify as a new technology introduction. My consulting practice over nearly two decades revealed to me a surprising lack of understanding of basic project management in many companies, much less the behaviors necessary for success with CCPM. 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. Yet CCPM appears to still be in the early adoption stage of a new technology.

Some have suggested that new ideas take a generation (i.e., twenty-five years) to be fully adopted. The old school has to pass on. On that scale, it seems to be doing pretty well. Perhaps it hasn't quite yet reached James's third phase, but Dr. Eliyahu Goldratt passed away in 2011, so perhaps it is a bit early for others to claim his accomplishment.

One thing is clear: You should now assume that your competitors are using it to improve their Throughput and quality and dramatically reduce project lead times. If you are not using it yet, that gap will likely continue to grow.

The most important thing I have learned through years of management experience is that the best managers keep things simple for the people performing the work. Dr. Goldratt's Theory of Constraints (TOC) addressed this at a global level, seeking simple solutions to seemingly complex organizational systems. I think it is important to always keep in mind that this element of TOC applies to each person in the system. You do not achieve better performance by modeling systems with increasingly more detailed project schedules or by having increasingly more complicated reporting systems. You can achieve huge performance improvements by enabling people to do the right work the right way. The right work is to focus work on the tasks that are necessary to complete the project on time. The right way is to focus that work on one task at a time until it is complete, and then move on to the next task. If you can do that, your systems will be Lean, and the flow of project completions will exceed your goals.

This third edition of *Critical Chain Project Management* expands on my integration of the improvement methodologies of professional Project Management, TOC, Lean, and Six Sigma. I sponsor synthesizing all of the alternative approaches to business improvement instead of defending one method as better than another. I value the areas in which the alternative business improvement methods seem to agree, because that supports their joint validity. I value more the areas that one set of ideas covers and another does not cover, because that expands our overall knowledge of the system. I most especially value the areas in which the improvement methods seem to conflict, because that is where I suspect the opportunities lie for additional breakthroughs. As all of the improvement methods have the same objectives, the apparent disconnects tell us that our picture of reality is not yet complete.

The third edition to *Critical Chain Project Management* provides solutions to reduce the waste caused by multitasking more so than the earlier editions. I now feel that multitasking is the enemy and in a variety of ways this enemy is winning by reducing productivity and quality across the globe at an increasing rate. The Critical Chain method described by Goldratt and in earlier editions of this book included some methods to combat one form of this enemy (overloading resources with multiple project tasks). Two new approaches offered in this edition provide help in two additional forms [overloading the system with too much project Work in Progress (WIP) and overloading resources with nonproject work].

This edition provides a clear solution to the problem that most companies cause their project systems to fail by strangling them with too much WIP, a term not usual to the project management field. The idea of controlling WIP to increase profitability has been understood in production for many years but is a new idea to project management and one not addressed strongly enough in the earlier editions of this book nor anywhere else in the project management literature. Dr. Goldratt understood it when designing the multiple project solution for Critical Chain (which I call Pipelining), but getting managers of project delivery systems to embrace this idea has been problematic. This third edition provides tested solutions for how to deal with excess project WIP.

This edition provides a solution to an additional problem not addressed in the earlier editions: in many project environments, the people who perform project tasks are also called upon to perform other types of work. This nonproject work frequently interferes with work on project tasks, usually by causing multitasking. It introduces mistakes into all work (also known as quality defects) and lengthens project duration. I recently found and have tested with clients a solution to this dilemma: Kanban. Although an old method in production and one of the up-and-coming Agile methods for software projects, this book is the first to introduce it for all types of project work and nonproject work. It is a means to control WIP at the working level.

This third edition also adds much new information, including a complete road map, on how to implement and sustain CCPM in a large organization. This edition reduces the information on other TOC tools because there are now better references for them.

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

I wish to acknowledge the special efforts of three people who have greatly contributed to my success with *Critical Chain Project Management*. The first is Dr. Eliyahu Goldratt, the Israeli physicist and production genius who put forth the idea of critical chain in the mid-1990s. He tirelessly presented it along with his other great ideas on system improvement around the world. He is the one true genius I have had the privilege of learning from directly and meeting with a few times. He inspired me and many others. Unfortunately Dr. Goldratt passed on in 2011 at too early an age. I sincerely appreciate what he gave me and so many others and miss his contributions to human knowledge and the human condition.

The second is Mr. Ronald Woehr of Orlando, Florida. He was instrumental in introducing and guiding the ongoing success of Critical Chain Project Management in one of the world's largest companies and has provided incredible assistance in the detailed editing of this work and has helped to draft some portions. He has been free with sharing his learning and success and concerns about all aspects of CCPM. He continues to challenge others with his leadership.

The third is my wife, Christina, who has supported me throughout my adult life, including the many hours spent at the computer preparing the three editions of this book. She continues to provide me daily lessons in how to contribute to the happiness of others in the world. This book could not exist without her.

CHAPTER 1

Quick Start

This chapter provides you with some things that you can do immediately to improve your personal and project performance before reading the rest of the book. It is for those who want practical advice without the theory. This chapter does not try to convince you of anything. I offer these items because I want your projects to succeed. My personal experience as a project and program manager and my work with dozens of organizations leads me to feel that this is the best advice I can share with you. You can choose to act on this advice now or after covering the more detailed "why" discussed in the following chapters. As you have invested your time and money in this book, I am confident that you will choose one of these approaches. However, I want to at least share the main "why" of this book now so that you start with some idea of where this is going and the reason behind the quick-start recommendations.

I have been a project and program manager for over 30 years and then a consultant to project management organizations for another 15 years. I was fortunate to learn how professional project managers plan and execute projects early in my career and the methods that I learned served me well, so well that I decided to become a consultant and teach them to others. Coaching a variety of organizations for the last 15 years provided me with a broad exposure to how many organizations plan and execute projects. Few exhibit the principles of effective project planning and delivery as defined by the international standard for project delivery: The Project Management Institute's *Guide to the Project Management Body of Knowledge*[1] is known as the PMBOK™ Guide (PMI, 2013). There are notable exceptions that do follow PMBOK-like practices, including major construction companies and large companies performing project work for the government.

Later I was fortunate to learn some very powerful ideas that apply to project management from Dr. Eliyahu Goldratt (Goldratt, 2007), in particular, his ideas about project management that he initially called Critical Chain. Later on, he and others adopted the terminology Critical Chain Project Management (CCPM), but not with the meaning I applied to it when I proposed it at a PMI conference in 1997. My meaning is the synthesis of the best parts of conventional project management as described in the PMBOK Guide with Goldratt's ideas about the criti-

1. Note that the project management body of knowledge itself comprises everything ever written on project management. The PMI document is a guide to the fraction of that knowledge that PMI determines to be in most common use.

cal chain. Today the PMBOK Guide and PMI's scheduling practice guide endorse Critical Chain scheduling (PMI, 2013), if not my full meaning of CCPM.

When planning project performance improvement, as I hope you are by reading this book, one starts with the problem or opportunity. I believe that CCPM both solves problems and exploits a huge opportunity to deliver quality project results much faster than you may think possible. My experience and study convince me that the main problem blocking businesses from greatly enhanced project delivery involves how they manage their people and projects.

One root cause of several project performance problems is that few organizations know how to create or use effective Project Plans. I mean Project Plans as contemplated by most of the project literature and described later in this book: more than just schedules. I was fortunate to learn the value of using Project Plans during project execution. The few organizations I find that create reasonable Project Plans sometimes do not use them effectively during execution. Nonuse leads to long-term degradation of the effort that they are willing to put into them in the first place. As the Project Plan content degrades over time, it becomes viewed as useless paperwork.

Perhaps the largest project execution problem is that most organizations allow or even encourage people to multitask on a daily or weekly basis (i.e., switch among multiple tasks before completing the first task they started). Many people believe that multitasking is a positive thing; they feel like they are being efficient. Some job postings suggest that it is a requirement for the job.

All current research demonstrates the error in believing multitasking is positive. Multitasking greatly increases the time that it takes to get all tasks done, causes a huge amount of waste reducing the total amount of work that gets done, and causes mistakes that then have to be corrected. The research also shows that those who think that they are best at multitasking usually are the least efficient at getting things done. They get the least done and make the most errors.

Electronic tools seem to contribute to the extent of multitasking. Trends in electronic media suggest multitasking is going to continue to get worse as more and more organizations and products compete for your attention and new people enter the workforce who have been brought up in the multitasking age. Management has to learn how to focus and coach their people to work effectively on project tasks. You already know that the only way to effectively complete a task with the highest quality that you can produce is to focus on that one task with no interruptions.

Organizations in which people work on both project tasks and other things (e.g., the holiday party, improvement projects, customer problems, factory support, and training) greatly exacerbate the multitasking problem. I call all of these other things *nonproject work*. Over my career, I have attended dozens of project management training sessions and have read hundreds of project management books and have not heard or seen one of them address the question of how to prioritize all work: project and nonproject. Although some companies have reasonable project execution systems that enable project task workers to mostly focus on one project task at a time (sometimes by assembling temporary project teams), I have not found a single one that provides a clear mechanism for prioritizing all work for focused execution by resources.

Although the problems related to multitasking have grown worse in recent years, they are not new and the solution is well known. At one time as a senior manager, I put much effort into learning personal "time management." I put "time management" in quotes because we cannot manage time; we can only manage how we use time to perform tasks. All of the time management training and books that I studied recommend the same thing. I recently heard a ditty that a person's grandmother repeated for him on a daily basis that provides that same solution:[2]

> If a task is once begun, never leave it till it's done.
> Be thy labor great or small, do it well or not at all.

That is not a bad summary of the key point of this book. We all know the answer to maintain our health and weight: eat less and exercise more. Some things are easier said than done. This is the case with focusing. Do you and the people with whom you work routinely focus to work on one task at a time until it is done while eschewing interruptions?

One opportunity to improve project performance includes assuring that the task on which your project resources work is the right task to focus on (i.e., is the task you select to focus on the one that you should do next to have the most positive effect on the company's performance?). You probably know that that the last task that you have been asked to add to your list is rarely the most important one for you to work on from the company's perspective, but how often do you see people switching to the task that was assigned to them last? I call this the LIFO problem, and correcting it provides an opportunity to greatly accelerate your project results.

Figure 1.1 illustrates a simplified logic to approach the problem and opportunity project performance in most organizations presents. You can read it from the bottom using the script, "IF <entity at tail of arrow> THEN <entity at head of arrow>." Thinking about it from the top down suggests that the bottom entity, management misunderstanding of the waste caused by multitasking, might be a root cause of project problems or missed project opportunities. My work over the last 15 years does not conflict with this idea.

Chapters 2 through 4 provide the reasons for CCPM, so if you are anxious to understand in detail what to do differently and why for a single project, you can jump to Chapter 5. If you are even more anxious to start a single project, you can start with Chapter 7. Chapter 8 guides you on executing multiple projects that share common resources.

However, if you want to jump in and do something now, read on.

1.1 Decide What Your Job Is

I hope that this book will be read and used by people in a variety of job roles. Section 9.1 describes the roles that I find in most organizations and one new role demanded by CCPM that most organizations lack (master scheduler). I expect that many of my readers will be project managers, supervisors, or schedulers. All of these roles influence how others do their work.

2. Although I am not a fan of rap, I must give credit where credit is due: rapper LL Cool J on *The Tonight Show with Jay Leno* in April 2013.

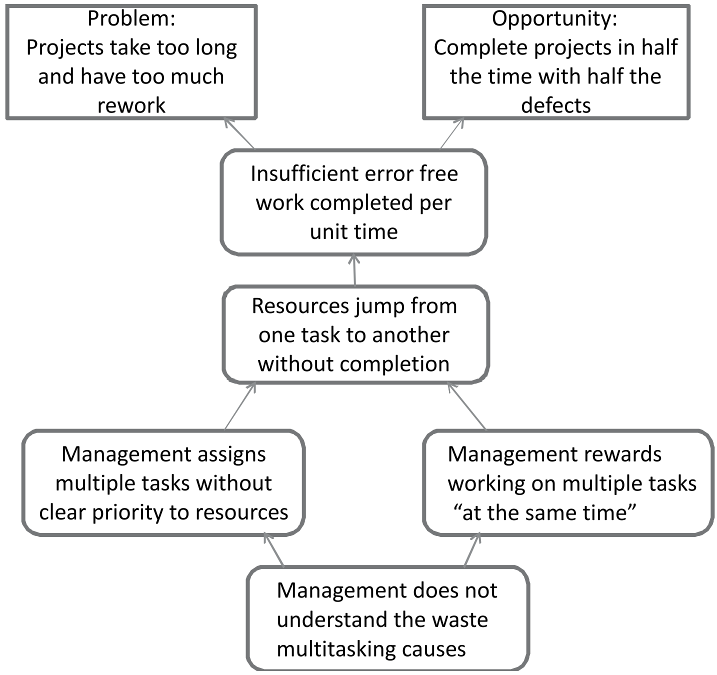


Figure 1.1 CCPM provides a solution to problems and opportunities.

Many managers have the mistaken impression that their job is to be the lead contributor to the product that their part of the organization delivers. A better idea, clarified by Mike Rother in Toyota Kata (Rother, 2010), is to consider the manager's job as the coach of his or her team leading them to continuously improve their work processes. The manager never should be out on the playing field. Many people (managers and workers alike) do not think that their job involves improving their work processes; they think that their job is simply to do the work.

Figure 1.2 illustrates a simple value stream for projects. It starts with a project Charter, the instrument that the organization uses to authorize starting the project planning phase of the process. Alas, because many organizations do not create Project Plans for their projects, many organizations jump from the charter to execution. In some cases, they jump into execution without a Charter. Most projects that proceed without following the value stream of Figure 1.2 end up not meeting one or more of a project's three major measures of success: full scope, on time, and within budget.

Adjust the amount of effort that you put into each of steps based on the value, complexity, and size of the project. Small projects require relatively little effort to create an effective Project Plan. Large, complex, high-value projects demand a very thorough project planning effort. All projects require that the primary focus be on effective execution to the plan.

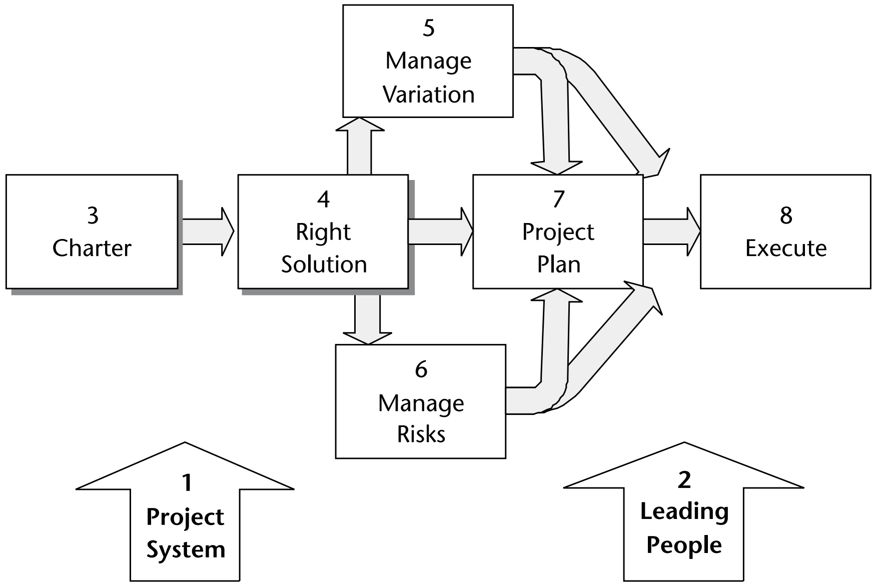


Figure 1.2 Example of the project delivery process value stream.

1.2 Use Appropriate Project Delivery Fundamentals

The first thing that you need to do to succeed with a project is to describe the result that you intend to create and plan how you are going to create that result. If you do not do that, chances are high that few will be happy with the project result. This applies to some degree for all projects small or large and simple to complex. The PMBOK Guide calls this a Project Plan, although some who work with Critical Chain prefer to call it a full kit.[3] Whatever you choose to call it, there are some key essentials that need to be in place for any project to succeed.

Figure 1.3 illustrates the major content areas of a full Project Plan. Most importantly, you have to define the end result of the project: the items under the Scope box in Figure 1.3. Defining the project scope helps you focus the team and to manage proposed changes to the project once you have started. Most successful project managers use a work breakdown structure (WBS) to organize the scope of a project and to assign responsibility to plan and perform the work. It is a deliverable-oriented representation of the complete project scope. The WBS is a simple tool. Create one as your first step for even the simplest of projects. If you need more information on the WBS, go to Section 5.4.2.1 or the *PMI Practice Standard for WBSs* (PMI, 2006). If you think you know it, read on.

Some confuse the WBS with the organization structure or with tools for budgeting and cost management on projects. Although you can use the WBS structure to organize budgets and collect cost, that is not its primary purpose. Please be clear on this: the WBS organizes the definition of the product the project is going to deliver, also known as the project deliverables (see Figure 1.4). The project

3. I only recently learned that the term “full kit” comes from the world of production management. This is not surprising as many people learned Critical Chain from Dr. Eliyahu Goldratt, who was a specialist in production management.

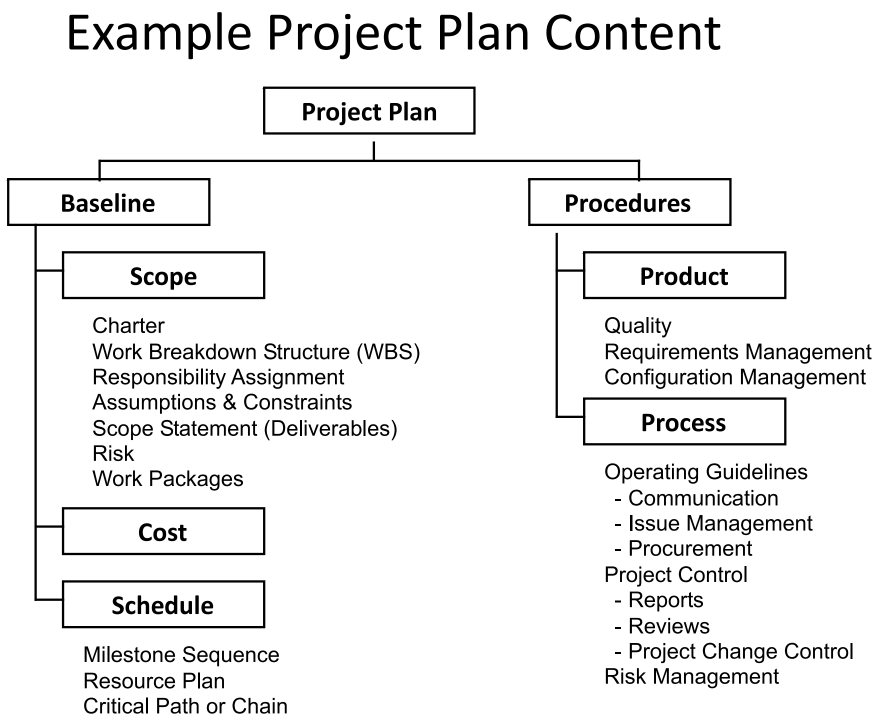


Figure 1.3 Scale Project Plan content essentials to your project.

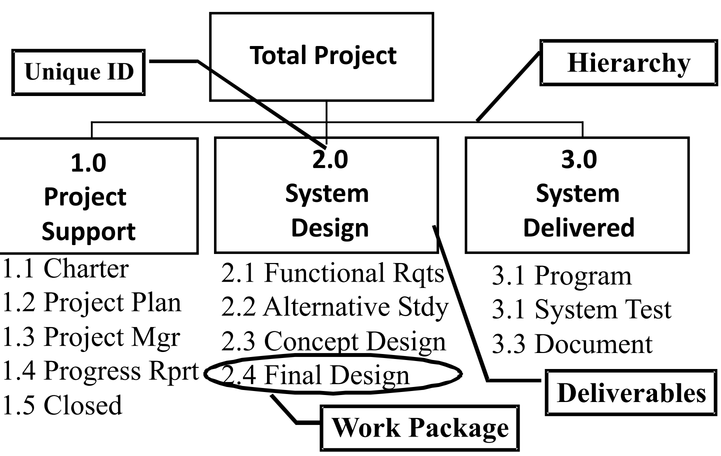


Figure 1.4 The WBS organizes the project scope and assigns responsibility.

management literature provides much useful information on the WBS if you are not familiar with it. You need to use it as your first quick step. If you need more information on it, jump to Section 5.4.2.1 before you come back here.

Record any assumptions that you need to develop the clear project scope and the way that you plan to go about creating the project results when you develop the WBS. Sometimes it helps to categorize two types of assumptions:

1. Those that relate to the product that you are producing, for example, assuming the technical direction that the product will take when your project includes analysis to decide which direction to take. This assumption may be necessary to decide what tasks to include in your schedule.
2. Those that relate to the project execution, for example, what work will be performed by in-house resources and what work will be performed by contractors. This assumption may be necessary to resource-load your schedule.

Update the assumptions whenever you need to complete the Project Plan, including adding assumptions as you develop the schedule. Assumption lists should not be overly long or complicated, but just enough to make it clear to your team and your customer what you believe to be true that might impact your project.

You need to establish how you are going to control the scope of the project. Your organization might provide a standard process for this or you might tailor one to your project. Your change control approach should include a simple form for people to write down their proposed changes to the project scope and the estimated potential impact on the scope, cost, and schedule for the project. Establish how you will approve proposed changes before implementing them and how you will ensure that everyone is always working to the most recently agreed-upon scope.

Your project team needs a schedule for your project. The schedule maps out the flow of work to create the project deliverables. The WBS provides no information on the flow of work. Many people confuse the schedule with a Project Plan. Although a schedule is a necessary part of a Project Plan, it is only one part.

Finally, the project team needs to know how they are going to work together to accomplish the project. The Project Plan needs to let your team know how you are going to status your schedule, how you are going to manage the issues that inevitably come up on projects, how you are going to make project decisions, and so forth. Unless there are documented universal processes for your projects, your Project Plan should clarify these processes. The right side of Figure 1.3 shows some of these processes. The most important processes involve communication. We now have a wealth of electronic tools that greatly enhance our ability to keep everyone on the same page and ensure that it is the right page (e.g., SharePoint sites, configuration management tools, and Web meetings). These tools are particularly important to support the growing trends of worldwide project teams.[4]

The above information provides the minimum set for any project. Larger projects may require much more. Even modest projects should have some kind of risk management approach. I will cover those topics in later chapters.

1.3 Enable Individual Task Focus

Management must help all workers to prioritize all their tasks (project and nonproject) on which to focus and work in sequence (avoiding multitasking). Then, when necessary, management must help workers resolve any blocking issues on the task on which they are working.

4. Some call these "virtual teams," but I think they are real teams simply displaced in space.

If you are the supervisor of one or more employees, you can start this initially by meeting with each member of your team and clarifying your expectations for focused work. Have the team members list everything on which they think they need to work and help them prioritize it. Make it clear to them that you want them to work on one task at a time and come to you to recheck the list as soon as they complete a task so that you can agree on what is next in case priorities have changed. That includes having them take any work requests that come to them from someone else to their supervisor for prioritization and perhaps reassignment.

If you are a project manager in an organization where the resources work for other managers, you have a more difficult problem. You have to engage the managers of the resources whom you plan to ask to act as above.

One way of doing that is to encourage the use of personal and team Kanban. The form of Kanban that I recommend is not what you may have heard about for just-in-time production. Although it uses the same basic idea (a Kanban or sign board) and the same principles that Ohno (Ohno, 1978) articulated, Anderson adapted it for knowledge work (Anderson, 2010). See Sections 3.6, 7.9, and 9.6 for more details on how apply this kind of Kanban for any kind of work.

The three most important rules of Kanban are visual control, limited WIP, and pull. (WIP is *work in progress,* not a standard project term but one you will hopefully understand fully by the end of this book.) Ohno developed six rules, which I will cover in Chapter 3, but I feel that these three are the most important ones. Visual control helps an organization learn to limit WIP to reduce multitasking and improve focus by working on one task at a time until it completes. The personal Kanban board can be at your workplace, but a team Kanban board must be on display in the workplace where the team members pass by frequently. Team Kanban operates the same as personal Kanban by adding identification of the resources performing the tasks. In both cases, the performing resource pulls tasks in to be worked on only when a previous task completes. Pull helps limit multitasking by putting the resource in control.

Figure 1.5 illustrates a current version of my personal Kanban board. It does not show the backlog of items from which I draw each week or the completed items archive. The small numbers above each section of a column provide a WIP limit. I allow myself only one work task at a time until it is done. I am more forgiving on the other categories, but I do not let them intrude on my work day and work week. When I complete a Doing task, I move it to the complete column and pull in another task from the ToDo column.

Group Kanban boards only cover project and nonproject work tasks with clear deliverables. Each person on a team can have his or her own personal Kanban board where the work task matches his or her task on the team board.

1.4 Develop and Manage to Project Schedules

As I worked with more organizations, I was surprised to find a lack of knowledge and skill for the creation and use of project schedules. Many organizations have none at all, just a project due date. Others have a few milestone dates and call that their schedule. Some confuse a budget with a schedule. Others apply project schedule tools, such as Microsoft Project, but use them incorrectly or ineffectively. For

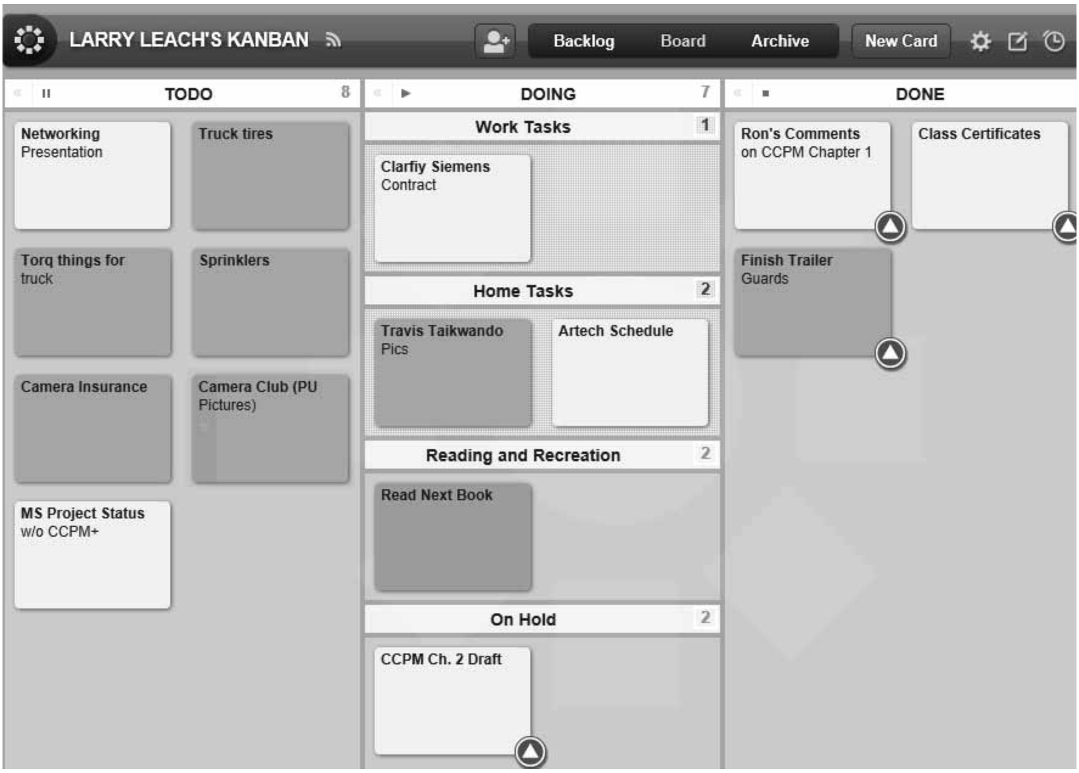


Figure 1.5 Example of a personal Kanban board.

example, they type in task start and finish dates instead of using task relationships to enable the software to calculate dates. Most do not consider resources, and many do not use the schedule as an ongoing tool to manage the project.

You must create workflow schedules for your projects. You create a schedule by developing the workflow (predecessor task to successor task arrows) to produce each deliverable in your WBS. Tasks must have relationships that connect the output (result of the work) of one task (the predecessor) as the input to downstream tasks (the successors). These links form chains of tasks and all of the chains of tasks must connect to a final milestone for the end of the project. You can do this with pencil and paper or with any of a multitude of scheduling software packages. The Project Management Institute provides a reasonable process for creating schedules (PMI, 2007).

Then you must track how the work flows relative to your schedule and take actions necessary to complete the project on the promised delivery date. A schedule is only useful if you use it on a daily basis to control your project. It is not something to be filed away.

So if you do not use schedules this way on your project, a quick-start item for you is to start doing so. You may need to get some training on how to use your software to create effective resource-leveled critical path or critical chain schedules for your projects, although I offer a caution on that. Much project software training will waste your time on how to use features that are not necessary and in some cases can even prevent you from using the schedule to control the flow of work on your projects. I will cover those dos and don'ts later on.

Figure 1.6 illustrates the Critical Chain process. It consists of creating and using a Critical Chain schedule, enabling tools, and execution behaviors. (I capitalize Critical Chain when referring to this process and use lowercase—critical chain—when referring to the schedule. Critical chain without capitalizationc contrasts with critical path scheduling.) You will do better with Critical Chain schedules than with Critical Path method (CPM) schedules for reasons that this book will go into later. Because a Critical Chain schedule is no harder to develop than an effective CPM schedule, you may as well create Critical Chain schedules at the beginning. The main things are:

1. Estimate your tasks with resources not doing multitasking.
2. Estimate the task durations with 50% likely duration. (This means that about half the time tasks should finish in less time than the duration in the schedule and half the time they should take longer. As a rule of thumb, when starting to implement CCPM, use half the duration you would normally use.)
3. Level the resources (i.e., adjust the project's demand for resources to the supply). All good project scheduling software provides this function. Your schedulers may need to learn how to load resources into a schedule and how to apply this function.

4 Place a buffer (tasks with no work) at the end of the project to account for the variation in the task chains. (As a rule of thumb, the buffer should be one-half the duration of the resource leveled project schedule from Step 5. This makes the project buffer one-third of the total project duration.) For a quick start, one buffer at the end of the project will suffice.

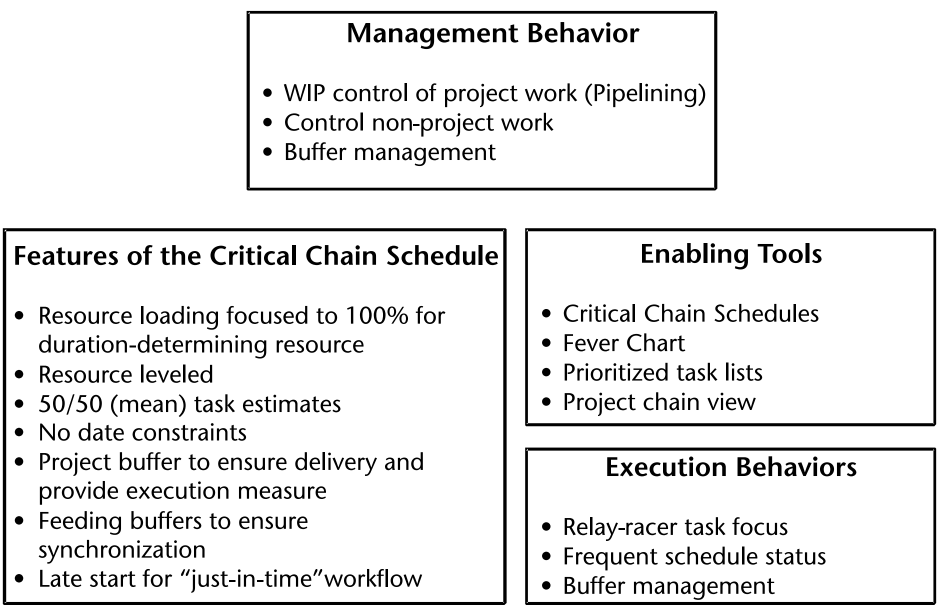


Figure 1.6 Key features of a Critical Chain process enable task focus.

Then use the schedule during execution:

1. Make the schedule you created above a baseline.
2. Status the tasks with actual start and finish dates as they start and finish. Also status tasks started and not yet finished on the day before your weekly project meeting with an estimate of the remaining duration required to complete the task.
3. In your weekly project meeting compare the statused schedule to your baseline schedule to see if you need to plan actions so the end of the last task will not exceed the project end date.

1.5 Control WIP at the Organizational Level

The first part of the following discussion, organization-level WIP control (also known as demand leveling), is for organizations doing more than one project at a time. This includes all organizations with which I have worked. I saved this as the penultimate item because most management teams find it is the hardest thing to do. They struggle with organization-level WIP control at the beginning of the transition to CCPM because they think that putting a project on hold means that it will finish later. They have not yet experienced the nonintuitive fact that working on fewer projects at a time means that more projects complete sooner. They have not experienced the tremendous acceleration in project completion that comes from reducing WIP and multitasking. You can tell them that it is a primary cause of the tremendous results achieved with Lean; they are not likely to believe you.

A first step to limiting organization-level WIP puts projects representing about one-half of the demand on your resources on hold and only restarting them as projects finish. Many organizations at the start find that they do not even know how many projects they have going and have only a general idea of the demand the projects put on their various resources. Do not agonize over this at first: make the best list you can of what is going on and put half of them on hold. Only restart projects as projects complete. Monitor workload and progress and make adjustments as necessary to enable the resources to focus on one project task at a time. You do not have to keep everyone busy; you need to ensure that no one is overloaded and plugging up throughput for everyone else.

The second step of organization-level WIP control addresses nonproject work. In the quick-start mode, the steps outlined in Section 1.2 may be the best you can accomplish. Over time, you need to develop a clear process for integrating the flow of nonproject work with the flow of project work at the task level so as to control WIP at the task level and cause a pull work environment where resources select and pull their next task on which to work only when they complete a task. Hopefully you can also create a new sensitivity to the effect that nonproject work can have on the flow of project work and organization profitability such that people become more cautious about initiating nonproject work and understand that their latest idea of something to do is not necessarily the highest priority for the company.

1.6 Summary

This chapter provides some steps that you can take right away without understanding the theory behind CCPM. The key points are:

- The most important thing that you can do is enable people to focus completely on a single project task and get it done before moving on to some other work: another project task or a nonproject work task. Needless to say, you can likewise help yourself most by modeling this discipline.
- Before starting project work, you need to ensure that all of the necessary information and materials are in place that will be needed to complete the project without delays. I consider the necessary information the primary elements of a Project Plan.
- If your organization performs more than one project at a time, or a very large project capable of overloading resources on its own, senior management needs to take action to control the overall WIP so as to not drive multitasking. This includes both project work and nonproject work.
- If people in your organization perform both project tasks and nonproject work tasks, you need a process to enable them to focus on one task at a time and tell them which one it should be.

Comparing the results of applying the Critical Chain practices to the way that many organizations perform work (i.e., with increasingly more multitasking) provides support for using the CCPM practices while we continue to review and improve it.

References

Anderson, David J. 2010. *KANBAN,* Sequim, WA: Blue Hole Press.

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

Ohno, Taiichi. 1978. *Toyota Production System*. Portland, OR: Productivity Press.

Project Management Institute (PMI). 2006. *Practice Standard for Work Breakdown Structures, Second Edition.* Upper Darby, PA: Project Management Institute.

Project Management Institute (PMI). 2007. *Practice Standard for Scheduling.* Upper Darby, PA: Project Management Institute.

Project Management Institute (PMI). 2013. *A Guide to the Project Management Body of Knowledge, Fifth Edition.* Upper Darby, PA: Project Management Institute.

Rother, Mike. 2010. *Toyota Kata.* New York: McGraw-Hill.

CHAPTER 2

Why Change How You Plan and Deliver Projects?

Projects continue to fail at an alarming rate. When I first did the research for this book in 1999, I found quantitative evaluations showing surprising rates of project failure. Those rates have not changed much at the time of this writing. As many as 30% of projects are cancelled before completion, wasting all the time, money, and effort spent on them. Surviving projects usually fail to deliver the full initial project scope or 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 often create personal and organizational pain and paperwork and often achieve little or negative impact on project performance. Improvements in the field of project management seem unable to keep 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.

This chapter provides the context for Critical Chain Project Management (CCPM), starting by defining the problem and using data to support the assertion that CCPM proves to be an effective solution for a wide range of project types and industries. The main points of this chapter are 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 3, which develops a firm basis for change in solution direction offered by CCPM.

As my experience grew with a wider variety and number of organizations performing projects, I came to understand a deeper cause for failure of projects today and for the seeming inability to improve project performance. While people continue to research the reasons for project failure and make new lists of what is wrong with the project management system or the managers who lead the system, they miss a deeper problem: managers' concept of what their role is in an organization. While this idea has been at the fringe of knowledge for a long time and somewhat addressed by the great management thinkers, such as Peter Drucker and W. Edwards Deming, no one articulated the idea as well for me as Mike Rother in his recent book *Toyota Kata* (Rother, 2010).

Rother chose the word kata to describe two major behavioral norms that his research uncovered within Toyota. The word kata comes from the martial arts and refers to a form that is so well practiced and learned that it happens without conscious thought. While most books on the Toyota Production System (TPS) and its derivative Lean Manufacturing describe a variety of tools and methods, Rother suggests that Toyota's success rests on a much deeper foundation of these management behaviors. The two behaviors differ significantly from those practiced by most Western management. Rother suggests successful application of the tools of Lean relies on two primary katas. The katas are: coaching and continuous improvement.

The coaching kata is just what you might think. Rother asserts that Toyota managers at all levels view themselves as coaches of the people who report to them. They are not drivers, judges, or the outstanding performer of the work that their organization produces. Thinking of them like the various coaches on a football team provides a reasonable analogy. I have had and observed many managers throughout my career in Western business and government organizations and only a few come to mind that behaved as if they were primarily the coach of the people who reported to them. What has your experience been?

Then, and perhaps this represents an even larger difference, the Toyota managers coach their teams to continuously improve their own work processes. The managers do not believe that they should be the ones to come up with or implement improvement solutions any more than a football coach thinks he should be on the field carrying the ball. Of course, when the coach has learned of a new process (e.g., Fosbury's flop in high jump where he broke tradition by going over the bar backwards facing the sky instead of foward, like all previous jumpers), it is essential to teach that new process to the team before you can expect them to improve on it. Most of you will be starting there with CCPM or PM 101.

The Project Management Institute's *Guide to the Project Management Body of Knowledge* (PMBOK™ Guide) (PMI, 2013) 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 wanted, when they wanted it, for a price they have agreed to, and having a project team that is happy about creating that success.

Chapters 3 through 5 refer to the existing project system. Although change is under way, most of the existing project management literature still primarily describes the Critical Path Method (CPM) to define a project schedule. The PMBOK™ Guide, PMI's *Practice Guide to Scheduling* (PMI, 2007), and most new project management books address Critical Chain. The PMBOK Guide's understanding seems to be getting it better in each edition. However, the *PMBOK Guide, Fifth Edition*, suggests that critical chain is an outgrowth of critical path. I do not think that is quite right. Asserting so is like claiming that Einstein's theory of relativity is an outgrowth of Newton's laws of motion. Instead, in both cases the new theory is more complete and embraces the old theory under special cases. Additionally, as I noted in Chapter 1, I consider CCPM to be much more than scheduling: it is more about management behavior and the impact on how people work. CCPM has to do with how management instinctively behaves relative to planning and executing projects much more than it has to do with just scheduling. In this way, CCPM is similar to Rother's katas: learned behaviors repeated without thinking about them.

Most commercial software claims to implement the CPM. Most of it can implement CPM when used properly. Most of the discussions that I have heard or read on CPM contain a hidden assumption that the estimated duration for the tasks that comprise a project schedule is deterministic. They do not mention the large variation that normally attends to most project tasks and how CPM users are supposed to account for it. Many discussions on making more accurate estimates suggest that the goal is to have a single-point estimate that the actual task durations achieve. This thinking demonstrates a misunderstanding of the meaning of accuracy. Accuracy means the range of the variation in the outcome. It does not mean that a single trial matches any single-point estimate.

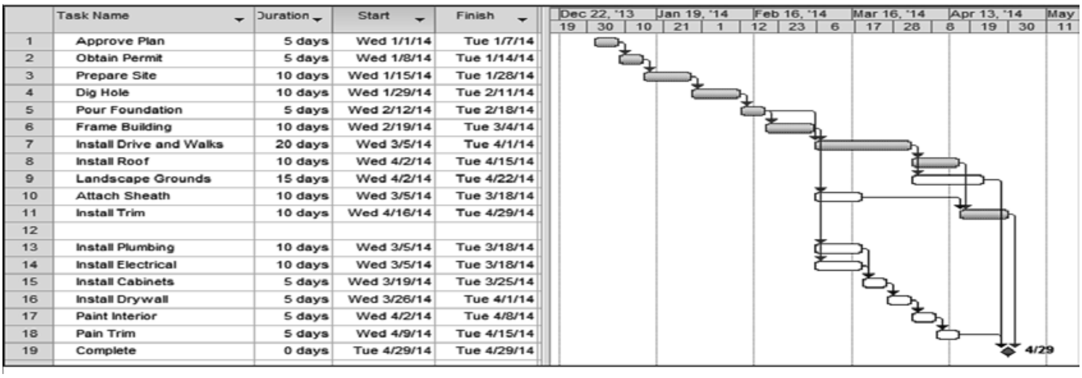
There have been people all along who understood the reality of variation in project tasks. Methods such as PERT and Monte Carlo have been proposed to deal with it. Users of those methods are the exception to the above assertions. What I have seen and heard of those methods lacks guidance on how to use them for daily execution management. For example, when using PERT or Monte Carlo, which schedule do you use to present the project and as the basis for decision-making on when and where to take action to recover schedule?[1] The methods that explicitly address variation in task performance are the exception in practice. There are a few notable exceptions where people or firms apply them quite well, most often on very large construction projects.

Instead of directly addressing project task variation, some of the discussions that I have seen on CPM, including the PMBOK Guide, describe methods to deal with uncertainty on projects through consideration of project risk. The PMBOK Guide and a practice standard also describe the earned value (EV) method of project measurement and control: another method filled with subliminal deterministic thinking (PMI, 2005). Many large projects use project risk management and the EV method, especially on projects performed for the U.S. government. Although not a specific point of guidance, most software and all of the applications that 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 2.1 illustrates a typical project schedule using this method. Few address how this method is the antithesis of the just-in-time approach adopted in manufacturing decades ago.

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.

I first learned of a system theory called the Theory of Constraints (TOC) in the mid-1990s. I read about it in Dr. Eliyahu Goldratt's book *The Goal* (Goldratt, 1984). 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 2.2 and describe the boxes as value-adding steps taking an input from an arrow and

1. There is one correct answer to this. One should control to a schedule comprised of mean task durations. The reason it that only the mean durations sum along a chain of chain of tasks to produce a mean estimate for the total chain duration. The sum of means is the mean of a sum. That is not true for other estimators such as the mode or median nor for optimistic or pessimistic estimates. I have never seen this answer given in discussions of such methods.



	Task Name	Duration	Start	Finish
1	Approve Plan	5 days	Wed 1/1/14	Tue 1/7/14
2	Obtain Permit	5 days	Wed 1/8/14	Tue 1/14/14
3	Prepare Site	10 days	Wed 1/15/14	Tue 1/28/14
4	Dig Hole	10 days	Wed 1/29/14	Tue 2/11/14
5	Pour Foundation	5 days	Wed 2/12/14	Tue 2/18/14
6	Frame Building	10 days	Wed 2/19/14	Tue 3/4/14
7	Install Drive and Walks	20 days	Wed 3/5/14	Tue 4/1/14
8	Install Roof	10 days	Wed 4/2/14	Tue 4/15/14
9	Landscape Grounds	15 days	Wed 4/2/14	Tue 4/22/14
10	Attach Sheath	10 days	Wed 3/5/14	Tue 3/18/14
11	Install Trim	10 days	Wed 4/16/14	Tue 4/29/14
12				
13	Install Plumbing	10 days	Wed 3/5/14	Tue 3/18/14
14	Install Electrical	10 days	Wed 3/5/14	Tue 3/18/14
15	Install Cabinets	5 days	Wed 3/19/14	Tue 3/25/14
16	Install Drywall	5 days	Wed 3/26/14	Tue 4/1/14
17	Paint Interior	5 days	Wed 4/2/14	Tue 4/8/14
18	Pain Trim	5 days	Wed 4/9/14	Tue 4/15/14
19	Complete	0 days	Tue 4/29/14	Tue 4/29/14

Figure 2.1 A typical CPM project schedule identifies the critical path and activity early and late start and finish dates. Most of the time, project schedules default to an early start schedule.

creating a value-added output that goes to the downstream tasks. I then ask people, "Which is this, a project or a production operation?" The reaction at first surprised me. Most people look puzzled at first. They do not respond immediately. Then one finally offers, "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 that we are going to explore in this book is the connection of dependent process steps that have variation in the time it takes to convert the input to the output of each step. Almost all such steps (project tasks or activities) have substantial variation in the time it takes to complete them.

The actual time on task, or touch time, in production operations is usually a very small part of the delivery time. However, when a machine works on a part in production it usually only works on one part at a 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 assumption in two ways: (1) multitasking, which causes idle time during the performance of project tasks; and (2) queue time, the time that project work spends waiting in queues.

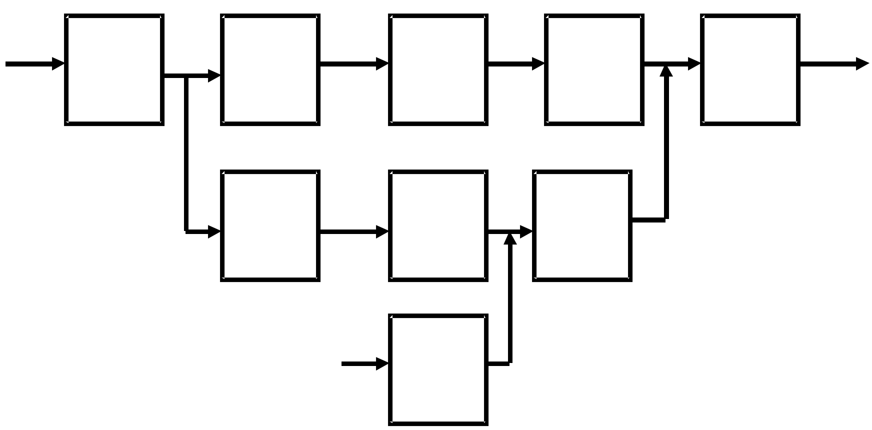


Figure 2.2 Is this a project or a production process?

2.1 Project Success

Successful projects meet the needs of everyone interested in the project: the stakeholders. All projects have a goal. Figure 2.3 illustrates that satisfying the goal normally requires satisfying three necessary conditions. The scope sets a minimum standard for the project results. Cost and schedule necessary conditions usually set maximums. Figure 2.3 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. Projects that take longer to complete cost more. Projects that cost more take longer to complete. The longer a project takes, the more opportunities exist to change the scope. The more changes to the scope, the more cost and schedule increase. Subsequent definition of the project system explores these relationships in detail.

2.2 Defining the Problem

Most scientists agree that precise definition of a problem is the most important step to a successful solution. Karl Popper (Popper, 1997), my favorite philosopher, notes that, "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 the system you presently use compared to CCPM. Hopefully you will agree that the problem definition "improve project success" is a bit too broad to guide developing a systematic effective solution.

2.2.1 How Good Is the Current Project System?

Ask yourself the following questions:

1. How often do you hear of projects taking longer than originally scheduled?
2. How often do you hear of projects completed much faster 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?

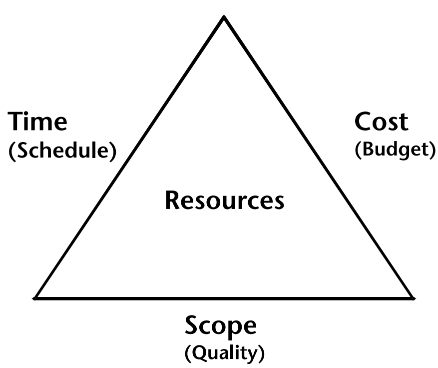


Figure 2.3 Satisfying the project goal requires three necessary conditions.

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

If your answers to these questions indicate other than full success on your projects, you have an opportunity to improve your project delivery process. Most organizations show opportunity relative to many of the questions. What if you could make a few simple changes that improved performance against all of these questions? I believe that CCPM can do that for you as it has done for many organizations. Although I see the changes as simple, the reality is that it depends on where you start. Many organizations that do projects have not defined project delivery processes that use the best features of professional project management. In those cases, I have found leading the organization to do more professional project management sometimes is not so simple.

Types of Projects

Table 2.1 illustrates one way of separating four categories of projects that measure success by completing a predefined scope on schedule and perhaps to a budget. The horizontal axis categorizes projects as 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 2.1 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 change while holding the schedule.

Table 2.1 Four Major Types of Projects Determine How You Should Plan

	Absolute Deadline	*ASAP*
External Customer	Type I: Proposal Event (e.g., Olympics) Contract with penalties	Type II: Construction Work system (e.g., ERP) Project input
Internal Customer	Type III: Regulatory Facility start-up Annual need (e.g., taxes)	Type IV: Improvement Product development Process improvement

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 and 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, budget) 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 the date. Finally, type IV projects 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.

There is at least one other type of project and one other kind of project failure worthy of mention. This type of project includes high-risk research endeavors where there is not an explicit path to the end result of the project. They are research projects. Some call such projects failures if they do not produce a desired research result (e.g., a new workable drug or breakthrough product design). For those types of projects, some counsel a portfolio management process that deliberately leads to a relatively large number of failures by this definition. The reason is that if they are not getting this type of failure, they are not pushing the technology far enough to develop really breakthrough projects. For example, Gartner (Gartner, 2013) suggested, "The accepted norm will be a 20 to 28 percent project failure rate as organizations are forced to accept increased risk to achieve desired returns." Others suggest an optimal failure rate for such projects is 50%. If such projects are carried out to do the research productively to quickly reach a definitive answer, I do not consider them project failures. They are trial failures that lead to overall program success.

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 the major projects that have suffered from the problem. Earlier versions of this book cataloged some that were recent memories at that time. As they are now a bit dated, I have deleted them. If this interests you, Google "project failures" and you will see an array of them. I recently got 165 million hits doing so. There is no indication that failure rates are decreasing.

Table 2.2 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 on "Why Projects Fail," and offer remedies to the various causes.

Table 2.2 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.

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. Starting with a GAO report from 1997 (GAO, 1997), here are just two quantitative examples:

> …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;

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

A 2003 update noted that despite sincere efforts to improve project performance (GAO, 2003):

> …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.

I checked again in 2013 and found (GAO, 2013):

> In response to GAO reports over the past few years on management weaknesses in major projects (i.e., those costing $750 million or more), the Department of Energy (DOE) has undertaken a number of reforms since March 2009... DOE's actions to improve project management are promising, but their impact on meeting cost and schedule targets is not yet clear. Because all ongoing major projects have been in construction for several years, neither EM nor NNSA has a major project that can demonstrate the impact of DOE's recent reforms.

For the second quantitative example, software projects seem particularly prone to failure. Some progress was reported following initial publication of the now-famous Chaos report in 1999 (Standish, 2004):

> 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%.

More recent updates show little progress. Project success of one-third remains a long way from my standard for success. How about yours?

The only common thread appears to be the project systems in actual use. Although there has been some progress for smaller software projects using Agile methods, larger projects tend to still use the present theory of the Critical Path method (CPM). (They may not all use it the same, 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:

- 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.

2.2.2 Some Companies Make a Lot of Money Running Projects

Despite these gloom-and-doom reports, many companies prosper in the business of running projects. What do these companies do that the losers are not doing? Much of the project literature would lead you to believe that they are the precious few who follow the PMBOK Guide in the most detail, and that all you have to do to join the successful is do 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. I have served on proposal evaluation teams where the difference between the winner and the second proposer was only 1% or 2%. For most companies, profit averages around 8%, so that is the difference between a successful company and one that goes out of business.

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. We rarely hear about the potential impact on the rest of the people in the company or how their suppliers get along. The model that we develop of the 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 it in the reasons projects fail, but perhaps not often enough. The answer is that 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 who attend the project management classes that I teach complain about scope creep. I tell them that 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. However, I control them by assuring the requestor that I will implement them immediately after they are approved by the project customer (even if it is the customer who is "directing" a change). I then only solicit that approval after rigorously estimating the scope, cost, schedule, and risk impact of each change, including the impact of cumulative small changes. You may find it 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. See Sections 6.11 and 9.7 for more information on formal change control. However, subsequent chapters reveal why change control 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 and dramatically reduces the number of changes.

2.3 Root Causes of the Problem

I learned the definition of a root cause long ago as: that which, if changed, will prevent recurrence of the problem. In this case, the problem is project failure and we need to know for sure what the cause is if we expect anything we do to have an effect on those failures.

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.

2.3.1 The TOC Method

The Avraham Y. Goldratt Institute asks project management students, "Why is it 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 caused the problem is outside the control of the Project Manager, and the cause is some type of unexpected event.

The problem statement that Goldratt proposed to develop Critical Chain 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 therefore theorized that the project systems failed to effectively manage uncertainty. Many TOC followers, including myself, then went on to draw cause-and-effect diagrams to test this assertion. We were usually able to support the assertion.

2.3.2 Project Management Literature

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 viewed by 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 itself. Here is my distillation of the lists I have seen, supplemented by my own experience:

1. Ineffective Project Manager:
   - a. Not assigned;
   - b. Untrained;
   - c. Two or more assigned Project Managers on one project;
   - d. Role as reporter (i.e., no authority).
2. Ineffective (or no) Project Plan:
   - a. Inadequate scope statement;
   - b. No execution guidelines:
     - i. Schedule use;
     - ii. Change control.
   - c. Schedule weaknesses:
     - i. No task relationships;
     - ii. No critical path/chain;
     - iii. No resources;
     - iv. Too detailed;
     - v. Level of effort (LOE) tasks, recurring tasks, and so forth included in schedule.
3. No status to schedule;
4. No control actions;
5. Ineffective change control;
6. Decisions:
   - a. Decisions not made;
   - b. Not assigned to one person;
   - c. No empowerment to decide and execute.
7. Multitasking execution:
   - a. Assigned to perform percents on different projects;
   - b. Interruptions with nonproject work;
   - c. Shifting project priority;
   - d. No task priority.

Note that none of the lists I have found in the project literature include Item 7: multitasking. I have come to believe it is by far the most important cause of most project failures. You might want to invert this list into a set of requirements for your improved management system.

Two assumptions underlie many of the evaluations leading to these lists:

1. *Project work is deterministic.* The evaluations address reality as if it were possible to get accurate or precise single-point estimates. Therefore, they assume that 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 to cause a particular failure. None of these studies question the effectiveness of the assumed system (which is often poorly defined in the studies themselves). None of these studies question the assumptions underlying the assumed effective system.

2.3.3 System Approach

One way to begin to understand project success or failure better is to look at the system to understand some of the assumptions that underlie it. Following Leopold (1933), 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 have more or less directly affected 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. An effective project control system;
5. Effective project execution;
6. An effective method to manage uncertainty.

Further expansion of an effective project control system leads to:

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.

Note that the first five of these influences might be combined as the management katas.

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., public).

Influences may affect one or more of the factors that more directly affect project success. Table 2.3 illustrates the relationship between the influences and the factors, and 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 and conflicts among all of the variables. For example, clients always want the highest possible quality while competitors force you to offer the lowest possible cost.[2] The project performance system is a complex system indeed. The complex relationships, 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 3, 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 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 policies of the company.

2.4 The Human Behavior Problem as Root Cause: Multitasking

Another way to think about the root causes of project failure is to consider it as a human behavior problem. Are there behavior patterns that, if changed, can dramatically improve project success? Because I ask this question, you might correctly assume my answer to be yes. My work over the last 15 years leads me to believe that the behavior that we call multitasking is the root cause of many problems observed in project performance. Fortunately, it is one we can do something about with CCPM.

2. Later on, I will show the win-win solution to this particular dilemma: lower cost caused by higher quality.

Table 2.3 Factors and Influences Affect Project Success

Factors That Determine Project Success	Influences on Project Success Factors						
	Internal Management	Measurement	Rewards	External Competitors	Suppliers	Client	Regulator
Right Problem	●					●	∇
Right Solution	●					●	∇
Effective Plan	●	●	∇				∇
Project Control System			∇				
Project Exection	●	●	∇				
Resource Quantity	●		∇				
Resource Skill	●		●	∇	∇		
Resource Behavior	●		●		∇	∇	
Work Processes	●				∇		
Changes	●				∇	●	∇
Uncertainty	●				∇		∇

● Significant Influence
∇ Some Influence

2.4.1 Multitasking

Supervisors typically support many projects at one time. All Project Managers want the supervisor's best person for their job, they want them as soon as they need them, and they want them full-time for the duration of their work on the project tasks. This leads to frequent conflicts over what project tasks people should work on. The usual solution is to assign people to work some fraction of their time on the tasks from each project. Using that strategy ensures that the organization gets the worst possible result for two reasons. The first reason is that multitasking causes all of the projects to take longer than they should, thus delaying the company benefit from those projects. There are two reasons for that. First, a person can only work on one task at a time. So if assigned to two tasks and switching back and forth, each task is idle half the time, causing projects to take twice as long. Multitasking among three tasks causes all the projects to take three times as long. Also, each task switch causes a loss while the person makes the physical or mental transition: pure waste. The second reason is that task switching causes quality defects.

One example of this that I saw early in my CCPM consulting was the repair of nuclear submarines at the Pearl Harbor Naval Shipyard. The Navy strove to make routine maintenance repairs on submarines within about 30 days. Before CCPM, they hardly ever succeeded in getting all the work done in that time. Crews assigned to work on tasks were often reassigned when something held up the work on that task (e.g., a missing part or instruction). This caused them to put that job back into a stable condition and move themselves and their tools to another location and have the necessary isolations made to start another job. They would not get far on that one until they found another cause of delay and had to repeat the job change process. This was to ensure that they always were working and had a job to which to charge their time. The CCPM discipline caused them to do a better job of ensuring that they had everything necessary before they started a job and redefined the supervisor's job to remove the blockage so they could finish what they were working on before moving to another job. The result was that nearly all submarines left on time with much more work done at less cost than happened previously. More surprising, quality defects, which they measured well, dropped by 50%. Productivity went up by 100%. Cost went down due to much less overtime. This was the result of eliminating the multitasking of work teams.

Multitasking's adverse effects apply to knowledge work as well as physical work. The term multitasking misleads because the conscious part of our brains can only attend to one conscious thought process at a time. Our brains are massive multitaskers in one sense (e.g., keeping us breathing, walking, talking, looking around, and so forth). We are able to do some conscious things, such as talking, along with other mostly unconscious things, such as walking or driving to a certain extent. Some parallel tasks do not interfere with one another. Others do.

When it comes to tasks that require conscious thinking, Medina (2008) noted, "To put it bluntly, research shows that we can't multitask…[we] must jump from one thing to the next." The thing that we call multitasking entails switching attention from one task to another before completing the first task. Each task switch requires both switching away from the task and switching back to it every time you allow your brain to make the switch. Human brains are prone to task switching, a fact exploited by commercial entities to serve themselves, not you.

We all have to perform different functions during the course of a day. The functions range from life necessities such as sleeping, eating, and exercising to different intellectual tasks. For our purpose with this book, each of those functions (e.g., sleeping) is a discrete task. You are not multitasking if you finish sleeping before moving on to eating. A bit more problematic are those tasks we decide to do over a period of time (e.g., doing a needlepoint, writing a book, or performing research for a thesis or work task). There is little question that I could have finished this book much faster if I spent all of my hours not devoted to life necessities on the book, but then I would have no income during that period, I would have lost prospects for future business, and I would have had some family issues with which to deal.

Of course, it makes sense to budget our time to things such as exercise, recreation, learning, family, community, and other worthy goals. My definition of multitasking does not preclude budgeting time for such activities, but rather encourages focusing on those activities during the times you have budgeted for them. So, for example, if you decided to budget time for working on your book, you do not jump to doing e-mail, answering the phone, reviewing the latest tweets, dusting the blinds, and so forth during those focused work periods.

Our brains can fatigue if we seek to focus attention on one item too long. Training and task content can affect how long we can focus without taking a break. Medina (p. 89) suggested that 10 minutes are about it for most students in a learning environment. Csikszentmihalyi (1990) asserted that "flow" experiences, where we completely focus on one thing, can last much longer.

Multitasking and the problems that it causes are not new. Cultivating the art of mindfulness goes back thousands of years. Much meditation works to increase mindfulness in order to correct the tendency of our minds to flitter from point to point and the consequent unhappiness it brings. Mindfulness focuses on quieting the micromultitasking. Increasing abundance and attractiveness of external stimuli causes our already flighty minds to latch onto one thought after another more rapidly and thereby increases the problems caused by multitasking. The large effects that one person can now have on others, such the drivers of large vehicles, are also new and increasing.

Conscious thinking is the most recent part of the human brain to evolve. It evolved on top of the more ancient systems. It was not a new design that started from scratch. Conscious attention receives all of the inputs from the senses plus all of the inputs the brain generates internally. While some of the sense inputs route around consciousness and go directly to action (you jump when you hear a sound before you experience the sound consciously), your conscious attention center treats input generated within your brain as if it were an external input. You can scare yourself in a dark, quiet room.

We could find no research to support positive consequence of the multitasking that often follows attention switching. A recent Stanford study which sought to find the positive effects from self-proclaimed effective multitaskers showed they are actually worse at task performance than those who can focus (Stanford, 2009). Stanford Researchers Eyal Ophir, Clifford Nass, and Anthony Wagner stated, "We kept looking for what they're better at, and we didn't find it."

Certain activities can surely profit from task switching. For example, I have to burn a set of CDs for training classes. Once set up, it takes about 3 minutes for a CD to burn. I can do other work while the CD is burning by interrupting that other work each time a CD finishes to replace it and start another one. The physical action to replace a CD and start another requires little cognitive effort. The other work, such as doing e-mail, might require attention. I have found that when the other work consists of small subtasks, such as doing e-mail, I can do this task switching just fine. I have also found that I cannot do other work requiring extended concentration, because the frequent interruptions of even the low cognitive attention task of changing a CD causes that other work to suffer, or the other work causes me to make mistakes on even the simple task of changing the CD.

You definitely can get more done by doing parallel work while some relatively longer-term automated process takes place, such as the dishwasher, washing machine, or dryer going through its nominally hour-long cycle. We all know we can walk and talk simultaneously, and maybe even chew gum. Some can rub their stomachs and pat their heads, and some cannot. Most of us are able to drive a car and listen to the radio or have a conversation with someone in the car with little apparent negative effect. Studies with teenage drivers show a marked increase in accident rates with multiple teenagers in a car. However, I have noticed that I (and many other drivers) reach over and turn down the radio when looking for a street or an address.

Brain research is now shedding light on why some tasks can be performed in parallel and some cannot. Put simply, if the tasks use different parts of the brain, they can be done simultaneously, whereas if they both demand certain parts of the brain, they conflict. Try this: cross your legs with the right leg on top and rotate your right foot clockwise. Now draw a six in the air with your right hand. You may have had trouble keeping your foot rotating clockwise because your brain was recruiting some of the same neurons to move your hand as it was using to move your foot and the directions conflicted. Tasks requiring conscious attention always conflict; we can only actually consciously focus on one task at a time (Medina, p. 84).

Some people have jobs that require many short duration tasks, for example, answering telephone inquiries. Switching from a completed task to another task is not what we consider as multitasking even if the tasks are very short duration and different from one another. My definition of multitasking is switching from an incomplete task.

It is possible to get stuck or bored while working on an intellectual task. In those cases, it often seems as though putting the task aside to return to it later can have positive effects. Sometimes with a task on which we are stuck, a new idea may emerge from our unconscious mind. This suggests that even though we consciously moved on to another task, our subconscious mind continued to work on the stuck task. That can indeed be a beneficial result from multitasking. We have found no research to support this claim, but personal experience suggests it may be true.

Likewise, when returning to a task with which we have previously become bored, we have found that we can feel reenergized and seemingly more productive. There are some facts of how the brain works that might lead us to expect that this might happen. Brain processes can become fatigued with use. Allowing them time to recover might indeed improve our performance. Once again, I could find no research to prove or disprove this idea.

Multitasking behavior shows some of the characteristics of addictive drugs. Multitasking sometimes feels good while you do it, but delivers devastating long-term consequences to your performance and health. Other times, multitasking greatly increases stress while you are doing it. The short-term effects of multitasking must provide positive consequences that reinforce the behavior; otherwise, we would not repeat the behavior. Long-term negative consequences have little effect on behavior because humans do not relate effects that are displaced in time from the effect. Many of the positive effects of not multitasking (i.e., focus) are long term.

Worse, because multitasking reduces your effectiveness and has no effect on the "incomings," the list of things on which you need to multitask grows like compounding interest. Simple awareness of a list of things that have to be done causes stress. Thus, the stress continues to increase and performance degrades further. It is a vicious cycle.

2.4.2 Multitasking Effects

Consider the research-proven effects of multitasking:

- Everything takes longer.
- You waste time with each switch of attention.
- You get less done.
- You make more mistakes.
- Your stress levels increase, leading back to the first point.
- Learning is reduced.
- Effects in organizations are exacerbated.

We will discuss each one.

Everything Takes Longer

First, when you multitask, you are actually switching your attention off of one task and onto another. While giving your attention to any one task, you are not working on the other tasks among which you are switching. Consider several tasks, each of which takes 1 hour of concentrated time. If you switch off to just one other task with equal time allocations to the two tasks, they both take 2 hours instead of 1 hour. If you switch among five tasks, they each take 5 hours or five times as long.

Here is a little exercise that you can try on your own to get the idea. You will need a timer for this. List numbers from 1 to 26 and letters from A to Z. Do it with multitasking the first time you try it, that is, write down a 1, then an A, then 2, then B, and so on. Check your total time. Then do it again but this time list all the numbers first and then list all of the letters. Record two times: the one for finishing the numbers and the one for finishing the letters. You will immediately see the obvious: although in this case the total time is not much different because the task switching loss is minimal, you got each individual task (i.e., listing numbers and listing letters) done in half the time. If someone is waiting for your result to do their work, just this can make a big difference.

Here are results reported by my colleague Ron Woehr of Orlando, Florida (personal communication):

> When I do this, I refer to this as two separate tasks. I record both the start and finish times for each task (both when multitasking and when focused) so that I can compare the elapsed time spent on each. After doing this with over 400 people, I can tell you the typical finding:
>
> 1. Elapsed time for each task when focusing is 33%–40% of the elapsed time when multitasking.
> 2. Total time to complete both tasks when focusing is 68%–80% the elapsed time when multitasking.
> 3. Around 23% of all participants make errors when multitasking.
>
> This leads to the following observations:
>
> 1. Requestors of the task results get a result in about 1/3 the time when resources can focus on the task.
> 2. Delaying the start of the second task assures the second task will be delivered earlier than if it were started immediately.
> 3. Many (and maybe most) nonconformances are not the result of a lack of intellect, knowledge, ability, or diligence on the part of the performer. They are the result of the way in which the performer is managed (i.e., asked to multitask).
> 4. The negative effects only compound if multitasking involves switching among more than two tasks and as the task complexity increases.
> 5. When integrating the two tasks into a single task (i.e., record the sequence 1A, 2B, 3C, 4D, ...), the multitasking time is much closer to the focused time. Apparently just separating the recorded numbers and letters by an inch or two exacerbates the task switching losses. Forced separation of the entries is a great example of a policy constraint.
> 6. The task touch time and task switching times are very nearly the same in this exercise. What happens if the touch time is long relative to the switching time?

> The answer depends on the duration of focused touch time and the amount of task switching time spent in a "now where was I?" mode. There is little "now where was I?" in the class exercise—until you get to the second half of the alphabet where most participants have more difficulty and where most of the nonconformances occur. So task switching at a rate that doesn't allow much time for focused effort after figuring out "now where was I?" is really detrimental to quality.

These results support the assertions I have made.

You Waste Time with Each Switch of Attention

Unfortunately, reality is much worse than just the time lost while tasks are idle. Each time you switch attention, there is dead time in which you produce nothing on either the task from which you switched or the task to which you switch. Studies show that the elapsed time to get back up to speed after switching off of a mentally demanding task can be on the order of 15 minutes or in some cases up to 40% of the time spent on task. This time adds to the overall duration of each task. So depending on how frequently you switch attention, a 1-hour task can take you 8 or 10 hours or more.

Research reveals that people have been aware of the time loss from multitasking for a long time. Some reference William James (1890) as the first to consider it. Arthur Jersild (1927) reported a pioneering study on task switching. His experiments were a little more complex than the exercise described above. He had subjects do something to a list of numbers (e.g., a mathematical operation). His experiments amply demonstrated the dead time and that there are a few factors that can influence it.

You Get Less Done

Multitasking tricks us. Most people feel like they are accomplishing many things at once while they are multitasking. They feel that they are getting more things done in a day. Nothing could be further from the truth. All tests show that people who focus on one task at a time and get it done get more done each hour, day, and week.

Being busy does not relate to getting more done. Quite the contrary, people who are busy by multitasking actually get less done. Have you ever waited at a traffic light behind people who do not go when the light turns green and then noticed that they are talking on their cell phone? Have you found yourself behind a very slow car and noticed that the driver was talking on his or her cell phone? Instead of getting multiple things done faster, multitasking makes everything take longer, in this case, the drive.

There can be certain kinds of tasks that do not require your attention so you can start them, leave and do something else, and come back later. Some household chores fit this model such as washing and drying clothes. I do not consider that multitasking, but rather consider it a task to load and start the washer, another task to unload the washer and load the dryer, and yet another task to unload the dryer and fold and store the clothes. Why is that not multitasking? Because you finished each operation as far as you can before you moved on to something else.

If you stopped halfway through loading the washer to answer the phone, that is multitasking.

You Make More Mistakes

Multitasking-caused mistakes can inconvenience others, put you at risk, and cost you your job and even your life. On October 21, 2009, Northwest Airlines Flight 188, an Airbus A320, flying to Minneapolis-St. Paul International Airport flew over the airport and continued to fly off course by 150 miles. The pilots did not respond to air traffic control. Fortunately, there were no other problems or impending collisions. Although the pilots originally stated that they were in an argument regarding airline policy and did not notice that they had flown off course, they later confessed to using their personal laptop computers at the time. The FAA revoked the pilots' commercial flying licenses.

Multitasking mistakes can hurt people. In May 2009, a Boston subway train crashed into another train at the station, injuring 49. The engineer was talking to his girlfriend on his cell phone at the time of the crash.

Multitasking mistakes kill people. At 4:22:23 PM on September 12, 2008, the Metrolink commuter train crashed into a Union Pacific freight locomotive in Chatsworth, California, killing 25 people and injuring 138. The National Transportation Board investigation showed that the engineer, who died in the crash, failed to react to signals. Investigation showed he was sent two text messages on his cell phone at 4:22:01 PM and 4:22:03 PM.

Hallinan (2009) provided an explanation of how multitasking caused Eastern Airlines flight 401 to plow into a swamp near Miami, killing all 99 people on board, including the multitasking crew in the cockpit. They were in the process of landing the airplane and decided to troubleshoot the reason that the landing gear light did not come on when it should have. Instead of first restabilizing the aircraft at a safe altitude, they shifted their attention to the light, allowing the airplane to fly into the ground.

Increasingly more evidence implicates multitasking in fatal and injury car crashes. A 2009 *New York Times* article noted, "the highway safety administration estimates that drivers using a hand-held device are at 1.3 times greater risk of a crash or near crash, and at three times the risk when dialing, compared with others who are simply driving." Other research shows that multitasking drivers are four times as likely to crash as people who are focused on driving. Research also shows that speaking on a hands-free phone poses the same risk as using a cell phone directly. Nonetheless, more states are passing legislation allowing the use of hands-free devices.

Medina (p. 87) noted studies that show multitasking causes tasks to take 50% longer and increases mistakes by 50%. We have already observed that multitasking larger tasks increases both the time by the number of tasks being switched among and by the waste time. That is, switching among four tasks (i.e., working 25% on each of four project tasks) means that each takes six times longer: four times longer because you are only working on the task 1 minute or hour or day out of four, and another 50% because of the task switching effect.

Your Stress Levels Increase

The American Institute of Stress (AIS, 2010) stated that the term "stress" was coined by Hans Seyle in 1936, who defined it as "the non-specific response of the body to any demand for change." Seyle apparently worked on the definition throughout his life, at one point posing what he thought might be a more useful definition as: "the rate of wear and tear on the body."

Most who deal with stress assert that the cost is huge. The largest cost is to individual health and happiness, which are difficult to express as a number. The monetary cost is huge, particularly in direct health-care costs and in the cost to business. AIS estimated the cost to U.S. businesses at over $300 billion due to "increased absenteeism, employee turnover, diminished productivity, medical, legal, and insurance expenses and Worker's Compensation payments." The AIS listed 50 "Common Signs and Symptoms of Stress" (see Table 2.4).

You can see from this list that some of the symptoms could be classified as immediate reactions to specific stressors, some as delayed reactions, and some as reactions to chronic stressors. Although many people view all stress as bad, those who work in the field recognize that stress can produce good effects such as aroused attention and thereby improve performance on tasks. Figure 2.4 illustrates the generally accepted view of the effect of stress on worker productivity.

Stress specialists note that the things that cause stress and the response to them vary widely among individuals. One person might completely ignore something that raises a high degree of stress in another, as measured by the physiological or psychological symptoms. That which causes stress to rise to a productive level in one individual may overwhelm another individual, causing his or her performance to plummet.

I do not propose that multitasking causes stress for all people. Nor do I claim that multitasking is the major cause of stress in the workplace. There is even some indication that a limited amount of multitasking reduces stress for some people. Multitasking may be as much an effect of stress as it a cause. There does appear to be a strong correlation between multitasking and stress (i.e., people report reduced stress when workplace multitasking is reduced).

Healy (2004) reported on research by University of Michigan psychologist David E. Meyer that suggested that prolonged multitasking may lead to shorter attention span, poorer judgment, and impaired memory. Healy also asserted that the "…epidemic of multi-tasking even is sending patients to doctors and therapists with complaints of depression, anxiety, forgetfulness and attention deficit disorder."

Learning Is Reduced

Multitasking has been shown to have a substantial negative effect on learning. Studies in classrooms where students were allowed to use their laptops during lectures clearly demonstrate that those who use the laptops, even if they use them strictly for subject-related uses, suffer a loss in retention of the material presented. Of course, those who use the laptops for other purposes during the class (e.g., e-mail or Facebook) suffer a significantly larger loss.

Table 2.4 AIS 50 Common Signs and Symptoms of Stress

1. Frequent headaches, jaw clenching, or pain	26. Insomnia, nightmares, disturbing dreams
2. Gritting, grinding teeth	27. Difficulty concentrating, racing thoughts
3. Stuttering or stammering	28. Trouble learning new information
4. Tremors, trembling of lips, hands	29. Forgetfulness, disorganization, confusion
5. Neck ache, back pain, muscle spasms	30. Difficulty in making decisions.
6. Light-headedness, faintness, dizziness	31. Feeling overloaded or overwhelmed.
7. Ringing, buzzing, or "popping" sounds	32. Frequent crying spells or suicidal thoughts
8. Frequent blushing, sweating	33. Feelings of loneliness or worthlessness
9. Cold or sweaty hands, feet	34. Little interest in appearance, punctuality
10. Dry mouth, problems swallowing	35. Nervous habits, fidgeting, foot-tapping
11. Frequent colds, infections, herpes sores	36. Increased frustration, irritability, edginess
12. Rashes, itching, hives, goose bumps	37. Overreaction to petty annoyances
13. Unexplained or frequent allergy attacks	38. Increased number of minor accidents
14. Heartburn, stomach pain, nausea	39. Obsessive or compulsive behavior
15. Excess belching, flatulence	40. Reduced work efficiency or productivity
16. Constipation, diarrhea	41. Lies or excuses to cover up poor work
17. Difficulty breathing, sighing	42. Rapid or mumbled speech
18. Sudden attacks of panic	43. Excessive defensiveness or suspiciousness
19. Chest pain, palpitations	44. Problems in communication or sharing
20. Frequent urination	45. Social withdrawal and isolation
21. Poor sexual desire or performance	46. Constant tiredness, weakness, fatigue
22. Excess anxiety, worry, guilt, nervousness	47. Frequent use of over-the-counter drugs
23. Increased anger, frustration, hostility	48. Weight gain or loss without diet
24. Depression, frequent or wild mood swings	49. Increased smoking, alcohol, or drug use
25. Increased or decreased appetite	50. Excessive gambling or impulse buying

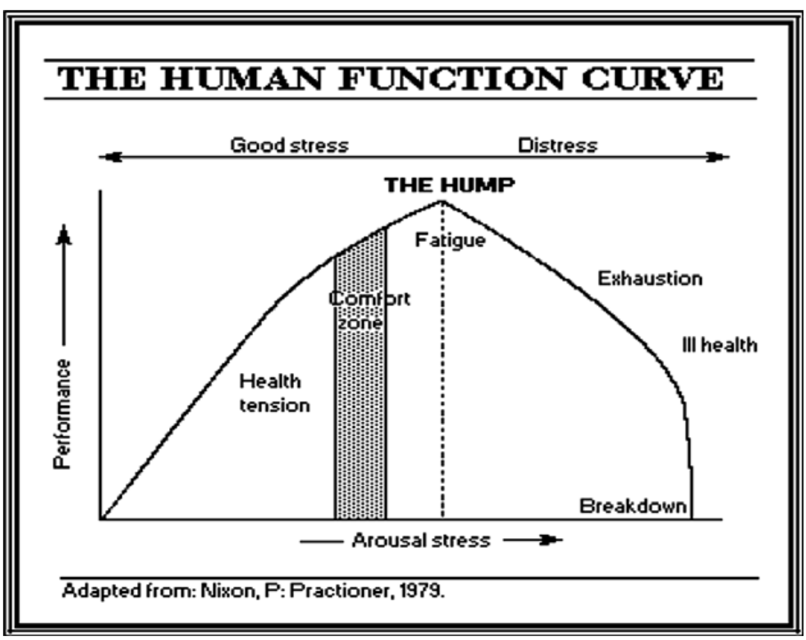


Figure 2.4 Human performance peaks at modest amounts of stress. (After: Nixon, 1979.)

Effects in Organizations Are Exacerbated

Most of the effects of multitasking propagate at least linearly into organizations. For example, the time wasted by individuals sums across all of the individuals in the organization. Some effects may be more than proportional, causing more total effect than the sum of the individual effects. For example, in organization workflows the mistakes made by one person are often not discovered by that person and passed on to others who may also pass on the mistakes and perhaps build on them. Research shows that the cost of correcting mistakes goes up rapidly as the discovery and correction of the mistake moves along a chain of process steps. The effects on learning and stress also no doubt translate to organizations, although there appears to be little research available on this.

Gonzalez and Mark (2004) studied information technology workers. They examined three different work roles: analysts, developers, and managers. They found no significant differences in fragmentation of work by these different roles. They noted, "Our study confirms…that information technology work is very fragmented…In a typical day, we found that people spend an average of three minutes working on any single event before switching to another event." They also noted, "People interrupt their work themselves (internal interruptions) about as much as they are interrupted by external influences."

Their definition of an interruption seems to correspond to the middle level of multitasking. They defined "work spheres" as corresponding to work tasks with a single definable deliverable (e.g., a project task). They found the average time knowledge workers spend on these work tasks is only about 12 minutes. That corresponds well with Medina's finding that people's concentration wanes after only about 10 minutes.

Follow-up work by Mark et al. (2005) clarified that the knowledge workers studied worked on an average of 12 work spheres or tasks in a day. This means that each takes at least a dozen times longer to complete and possibly much more. Their study showed that people switched working spheres on their own. They did not need an interruption to multitask; they interrupted themselves. They observed that, despite this, most of the workers reported that they preferred to complete one task before moving onto another. This leads directly to the CCPM solution.

2.5 Right Solution

People have posed many solutions to improve project management over the last 40 years. They attempt to better meet the customer needs on time and at or under budget. Solutions generally trend in the direction of providing 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.

Dr. Goldratt began *Critical Chain* (Goldratt, 1997) with a discussion of a company wanting to reduce the time on critical development projects. The company 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 from all of these changes, the consultants concluded that it would save: "Maybe five percent. Maybe not even that."

2.5.1 Do More Better

Earned value and derivative cost schedule control systems (CSCS or CS squared) (Lambert, 1993) frequently increase the detail of project schedules 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 goes into 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 20 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 plan a project crept up to almost 7 years. This does not include the time to do the project. That is, they have built in 7 years of planning time before start of any project. There are engineering studies, conceptual design reports, independent cost estimates, and validated CSCS. Yet the cost and schedule of projects continue to rise, and more projects fail to meet technical requirements. In one case, the government agency canceled a project after spending over a billion dollars on it. Other projects are decades late.

One study showed that their buildings 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 crises, where they would "rebaseline," yielding new cost and schedule estimates several times (usually three or more times) the original estimates. Project Managers cancelled larger projects because the need was gone before the project was over, or because the newly projected cost and schedule changed the cost-benefit equation to where the project no longer made sense. This is the problem that Project Managers were trying to solve in the first place. Is the world changing that much? Alternatively, 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 the scope, schedule, and cost, you must define those requirements accurately. To define requirements accurately, you must add detail to your project schedules, because previous projects failed to deliver at the present level of detail. This logic seems to make sense and to be in line with literature that attributes project failure to inadequate requirements or insufficient detail in the project schedule.

The do-more-better approach frequently leads to project schedules with tens of thousands of activities. I have worked with clients who were rather proud of the fact that their project schedule contained over 30,000 activities.

Later on, I will go into the factors that will help you determine the size of activities that you should have in your project. In some cases, this may lead to many thousands of activities for very large (multibillion-dollar) projects.

To put it in perspective, consider a much more modest project schedule, 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 happy as a clam 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 much bigger than 1%, is that they miss 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 streetlight because, "I 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 you have, rather than breaking the defined activities into greater detail.

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

1. Theory of knowledge: understanding that one or more successful cases do not prove a theory (discussed in Chapter 3).
2. Their environment may have had very poor practices to begin with, and any degree of discipline is likely to cause an improvement.
3. Regression to the mean: understanding that a particularly bad performance in likely to be followed by a better performance.
4. The Hawthorne effect: a psychological effect where 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 test.

2.5.2 Variation and Uncertainty

Everyone knows that project tasks have a certain amount of inherent variation and they also are uncertain 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. To complete the project successfully, you must account for this variation and uncertainty. People's ability to estimate off the cuff depends on a number of factors. There is substantial evidence to indicate that people tend towards overconfidence in their belief in the accuracy of their estimates (Kahneman et al., 1982). The data that I have found suggests it is likely that most project task durations have a two-sigma range (i.e., about two-thirds of the tasks complete within the range) of more than ±50%.

I have people estimate a very simple task as part of my 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 2.5 illustrates combined results from several of these exercises.

Figure 2.6 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 accuracy of zero. An estimate with no effort at all should have an accuracy of at least 100% on the down side and could be orders of magnitude (hundreds of percent) too low. The curve illustrates that the accuracy should generally improve as more effort is put towards 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 the next chapter, 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 2.6, divided by the vertical 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 estimating 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 that you have identified in your project schedule identify deliverables (clean hand-offs from one primary resource type to the next), the effect on overall schedule uncertainty that will obtain from subdividing tasks in your project schedule 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

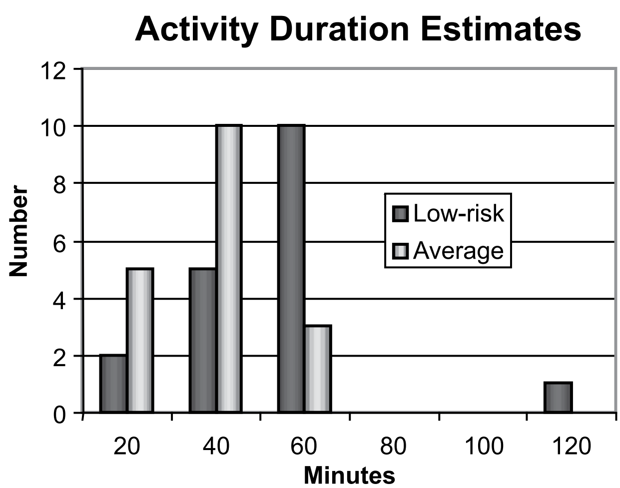


Figure 2.5 Estimate uncertainty for a very simple project task illustrates typical range of real uncertainty.

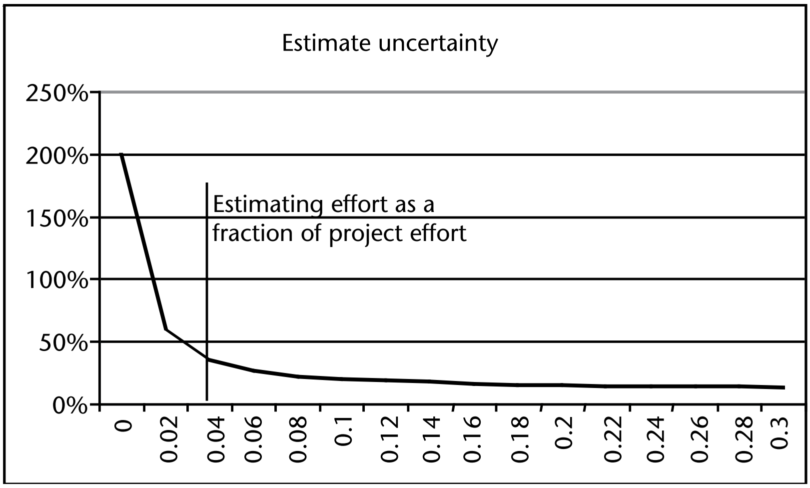


Figure 2.6 Estimate accuracy generally increases with the effort applied to the estimate, up to a limit determined by the process involving the subject of the estimate.

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 2.6, 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 schedule increases the number of potential task connections much faster than the number of tasks you add. For example, if you add one task to a plan with 100 tasks, you only add one 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.

A cause you might deduce from project failure due to the alleged causes of project failure posed so far is that uncertainty causes projects to fail. If this were the cause, you should predict that all projects with uncertainty 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. In *Critical Chain*, Dr. Goldratt describes one airplane project that defies this prediction. The designers developed an airplane with unprecedented capabilities in 8 months instead of the 10 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 and, 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.) Although 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 projects to fail. 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.

2.6 Right Execution

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

Dr. Goldratt noted in his pamphlet, "My Saga to Improve Production," (Goldratt, 1994) that:

> 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 performance of the project system. 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. Many interpret doing better as developing 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 11 provides an effective plan to implement the Critical Chain project system.

2.7 Success with Critical Chain

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 were 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, and that the theory can explain why. These benefits [compared to the present Critical Path Method (CPM) theory] are:

- Improved project success:
  - Projects complete on time all the time;
  - Projects deliver full scope;
  - Project cost under budget;
  - Improved market position and business growth.
- Reduced project duration:
  - Projects complete in one-half the time (or less) of previous similar projects;
  - Individual project schedules reduce the scheduled duration by at least 25%;
  - Project durations for projects in a multiple project environment reduce by larger amounts;
  - Reduced project changes;
  - Early returns for commercial projects;
  - Reduced payback periods for investment projects;
- Increased project team satisfaction:
  - Reduced confusion from multitasking;
  - Ability to focus on one task at a time;
  - Reduced changes;
  - Reduced rework;
  - Reduced pressure from multiple Project Managers;
  - Elimination of win-lose task completion (date-driven task pressure);
  - Individuals use buffer report to decide their own task priority;
  - Reduced insertion of new priority tasks;
  - Simplified project measurement;
  - Quick and easy plan status;
  - Real-time project status, no need to wait for financial reports;
  - Status provides immediate focus by buffer, chain, and task;
  - Buffer report defines decisions;

  - Buffer reporting focuses decisions on management priorities (reflected in the buffers by staggering project start).
- Simplified Project Management:
  - Clear focus for project manager (critical chain, reduced early start);
  - Simplified project schedules reduce schedule development and maintenance effort;
  - Simplified project status reporting;
  - Measurement guides decisions to plan or act;
  - Measurement guides decisions on resource priorities.
- Increased project throughput with same resource:
  - Reduced resource demand conflicts;
  - More projects completed faster for the same level of resources;
  - Less need to hire new critical resources;
  - Less delay due to resources;
  - Improved project cash flow;
  - Improved return on investment (ROI).

Evidence of other users often gives people confidence to try new ideas. The present Critical Path Method (CPM) project paradigm has been in force for over 60 years, making change very hard for many people to accept. 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.)

Rather than a long of list of specific instances of project success that will become dated, I can recommend some Web sites for you to go to for a wealth of analytical and anecdotal data on success with Critical Chain. The first one presents many videos of their clients providing details of their success:

- www.Realization.com;
- www.Goldrattconsulting.com;
- www.Goldratt.com;
- www.Prochain.com.

I invite you to look for more by Googling variations of "success with Critical Chain." I got 805,000 results. Almost everyone wants to find someone "just like us" who succeeded before they try it. Of course, they also claim to be the leaders in their market.

Despite that, I will offer some specific examples without the names of for-profit companies because many of them consider CCPM a competitive advantage. The U.S. Air Force and Navy have had huge successes. The Air Force has increased the throughput on major airframe maintenance work to keep more helicopters and airplanes in the air rather than waiting on maintenance. The value of those, in effect, newly created operational machines ranges to billions of dollars. The U.S. Navy and U.S. Coast Guard completed ship repairs in record times also increasing the availability of those very expensive machines, effectively creating new ships out of nothing. Aerospace companies have reduced cycle time by half and increased throughput by 50% and more on engineering projects. Companies that make consumer goods have increased throughput of new product developments by up to several hundred percent. The products range from relatively simple consumer goods to highly complex machinery for the industrial and government markets. Companies that make very large industrial equipment such as the electrical generators for central station power plants have increased engineering project throughput by over 100% and reduced cycle time by half. Oil refineries and chemical processing plants have greatly reduced maintenance shutdown time.

One company noted above, Realization, Inc., led me to the idea of capturing Critical Chain Project Management in terms of three new rules. I liked the idea of three rules because that is the number people seem most able to recall. I changed the specific definitions of the rules from those they proposed. I believe my rules to be a bit more project management oriented and to communicate better. So far, my students seem to agree.

2.8 Three New Rules

The three new rules I use to capture the essence of CCPM are:

1. Focus;
2. Buffer;
3. Pipeline.

Because rules 2 and 3 are there to enable rule 1, if you would like to simplify further to one rule, it is the first one: focus. CCPM is all about enabling people to focus on one task at a time and successfully complete it. Of course, it helps if it is the right task to focus on. Management's job is to make that possible (identify the right task and insulate resources from pressure to multitask) and to resolve issues (such as necessary decisions or missing task inputs) that make completing a task as soon as possible difficult.

Buffering is the mechanism CCPM uses to manage variation in project tasks. We will cover it much more in future chapters. I feel it is the primary contribution that Dr. Goldratt made to successful project management. Although because of his now-famous book *Critical Chain* (Goldratt, 1997) many think that Goldratt invented the idea of the critical chain, he was not the first to propose that the longest sequence through the project should consider both resources and task logic. Wiest (1963) proposed the idea under a different name: the critical sequence. Although he did not include the idea of managing variation and does not appear to have developed the idea into an execution methodology as has happened with Goldratt's critical chain, he should be given the credit for the critical sequence (also known as the resource-leveled critical path) idea.

Pipelining is the equivalent to demand leveling in Lean manufacturing. The idea is to level the demand for resources to work so they are not overloaded. The reason is because overload of resources leads to long queuing delays and consequent increases in Work in Process (WIP) inventory. The idea of WIP is foreign to project management. I will show why it is as critical to management of multiple projects as it is to manufacturing. It is something that managers of organizations with multiple projects should take primary responsibility for. Most do not.

2.9 Summary

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

- Projects success rate using the existing critical path paradigm (or method: 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 of working harder with the old system: the do-more-better approach. This does not seem to address the right problem.
- Part of the problem relates to Western management's understanding of their primary function. It should be to coach workers to continuously improve the production process. The other stuff they do should come in at much lower priority if at all.
- The original cause of project failure posed by Dr. Goldratt, and leading to Critical Chain, was "Management's failure to manage variation."
- A major cause of ineffective project performance in organizations today is the multitasking of project task workers: switching from one incomplete project task to another task. Eliminating this multitasking offers a huge opportunity for performance improvement.
- Improvements using the do-more-better approaches 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 and fails to enable effective human behavior.
- The right solution requires a project system that delivers 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.
- Three new rules of project management summarize CCPM as it has developed from Goldratt's Critical Chain: (1) focus efforts on the right tasks, (2) buffer to manage variation during execution, and (3) pipeline: control the overall demand on the project system resources to increase velocity and throughput.

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

References

AIS. (2010). AIS The American Institute of Stress. http://www.stress.org/topic-definition-stress.htm.

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

Csikszentmihalyi, M. (1990). *Flow*. New York: HarperPerennial.

Gartner. (2013). "Gartner Says Smart Organizations Will Embrace Fast and Frequent Project Failure in Their Quest for Agility." http://www.4-traders.com/GARTNER-INC-12710/news/Gartner-Inc-Gartner-Says-Smart-Organizations-Will-Embrace-Fast-and-Frequent-Project-Failure-in-The-16833524/2013/

GAO. (1997). Department of Energy: Improving Management of Major System Acquisitions, GAO/T-RCED-97-92.

GAO. (2003). Status of Contract and Project Management Reforms. Statement of Robin M. Nazzaro, Director Natural Resources and Environment. GAO-03-570T.

GAO. (2013). "Concerns with Major Construction Projects at the Office of Environmental Management and NNSA," GAO-13-484T.

Goldratt, Eliyahu M. (1984). *The Goal*, Great Barrington, MA: North River Press.

Goldratt, Eliyahu. (1994). "My Saga to Improve Production," New Haven, CT: Avraham Y. Goldratt Institute.

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

Gonzales, V., and Mark, G. (2004). "Constant, Constant Multi-Tasking Craziness: Managing Multiple Working Spheres." University of California, Irvine.

Hallinan, J. (2009). *Why We Make Mistakes*. New York: Broadway Books.

Healy, M. (2004). "Multitasking: Shortchanging Brains, Lives." *Los Angeles Times*.

James, W. (1890). *The Principles of Psychology. Vol. 1*. New York: Henry Holt.

Jirsild, A. (1927). "Mental Set and Shift." *Archives of Psychology*. No. 89.

Kahneman, D., P. Dlovic, and A. Tversky. (1982). *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge, U.K.: Cambridge University Press.

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

Leopold, Aldo. (1933). *Game Management*. Madison, WI: University of Wisconsin Press.

Mark, G., Gonzales, V., and Harris, J. (2005). *No Task Left Behind? Examining the Nature of Fragmented Work*. Portland, Victoria, Australia: ACM.

Medina, J. (2008). *Brain Rules*. Seattle, WA: Pear Press.

New York Times. (2009). "Drivers and Legislators Dismiss Cellphone Risks." http://www.nytimes.com/2009/07/19/technology/19distracted.html?pagewanted=all. Accessed December 28, 2009.

Nixon, P. (1979.) Practitioner.

PMI. (2013). *A Guide to the Project Management Body of Knowledge, Fifth Edition*. Upper Darby, PA: Project Management Institute.

PMI. (2005). *Project Management Institute, Practice Standard for Earned Value Project Management*. Upper Darby, PA: Project Management Institute.

PMI. (2007). *Project Management Institute, Practice Standard for Scheduling*. Upper Darby, PA: Project Management Institute.

Popper, K. R. (1997). *Objective Knowledge: An Evolutionary Approach*. Oxford, U.K: Clarendon Press. p. 144.

Rother, M. (2010). *Toyota Kata*. New York: McGraw-Hill.

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

Stanford. (2009). "Media Multitaskers Pay Mental Prices, Stanford Study Shows." http://news.stanford.edu/news/2009/august24/multitask-research-study-082409.html. Accessed December 28, 2009.

Weist, J. (1963). "Some Properties of Schedules for Large Projects with Limited Resources." University of California, Los Angeles.

CHAPTER 3

The Synthesis of TOC and PMBOK, Considering Lean and Six Sigma

This book approaches the problem of improving project management from the perspective of synthesizing two major domains of knowledge: the project management body of knowledge (PMBOK) and the Theory of Constraints (TOC). I consider this synthesis with perspectives from two other knowledge areas: Lean Manufacturing and Six Sigma. Figure 3.1 illustrates the synthesis. This chapter addresses each of these PMBOK™ Guide[1] (PMI, 2013) knowledge areas in order and illustrates their relationship to the overall Critical Chain Project Management (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. 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 3.1 illustrates how the multiple perspectives on the project system might look at problems in project performance. The PMBOK Guide perspective compares actual project system performance to the PMBOK Guide model, which it assumes correct. Therefore, the PMBOK Guide perspective is unlikely to blame elements of the PMBOK Guide project system as the cause of the problems. The PMBOK Guide perspective is more likely to blame performance problems on failure to execute properly its (assumed) effective system. A natural consequence of solutions based on the PMBOK Guide perspective is to try to "do more better." This is indeed the nature of much of the project management literature, as described above. Dr. W. Edwards Deming noted that you should not expect significant system changes to come from within the system.

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 effectively execute projects 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

1. PMI differentiates between their publication, A Guide to the Project Management Body of Knowledge (PMI 2013), PMBOK Guide, and the actual knowledge base including everything ever published on project management, or the PMBOK. I try to follow that guidance.

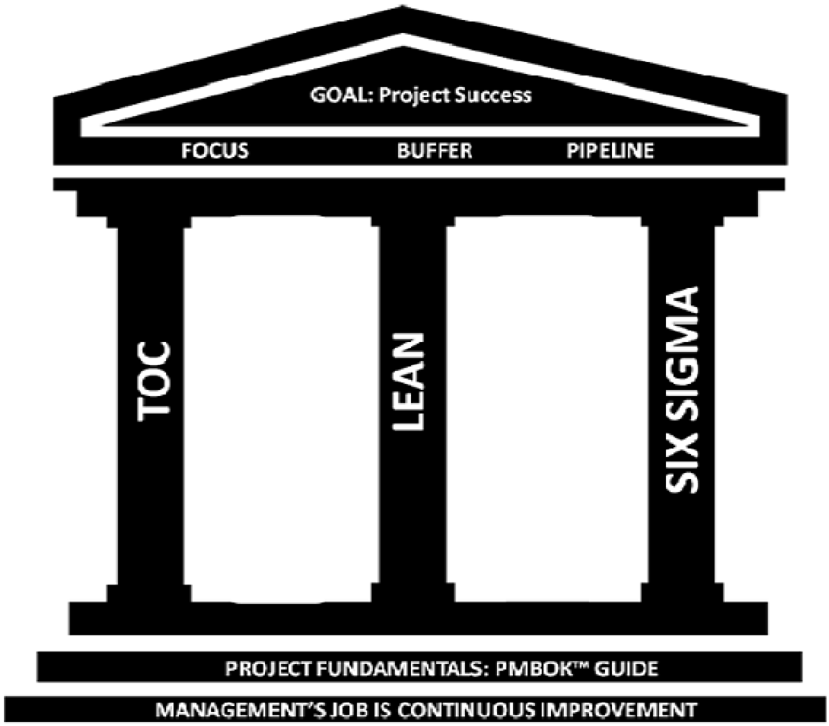


Figure 3.1 Multiple knowledge areas increase perspective on the Project System PMBOK™ Guide Perspective.

have contributed to two of the last three versions. I feel that the fifth edition continues the cycle of improvement. I view this book as part of my ongoing effort to improve project management systems and I expect the PMBOK (not just the Guide) to embody some of the methods that I describe here as they become more common. Since the first edition of this work, I have seen substantive progress in that direction. The PMBOK Guide (since its third edition) and PMI's scheduling practice guide (from its first edition) both now endorse Critical Chain scheduling, if not all the principles of CCPM. I also believe many of the elements of the PMBOK Guide are necessary conditions to successfully deploy CCPM and will identify them in the appropriate places.

3.1 Improvement Perspectives

Six Sigma and its predecessor, Total Quality Management (TQM), seek to continually improve every process, the latter 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.

3.2 TOC Perspective

The TOC perspective identifies the system constraint, and works to improve its throughput. It provides a system view of projects, a specific theory to predict project performance, and the impact of changes to the system. The TOC 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 the individual project tasks interrelate.

3.3 Project Management Body of Knowledge (PMBOK)

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

Personal computers 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 50 years ago.

Figure 3.2 illustrates the related knowledge areas identified in the PMBOK Guide. This text focuses on and proposes changes to four of the project management knowledge areas to impact the necessary conditions for project success. These

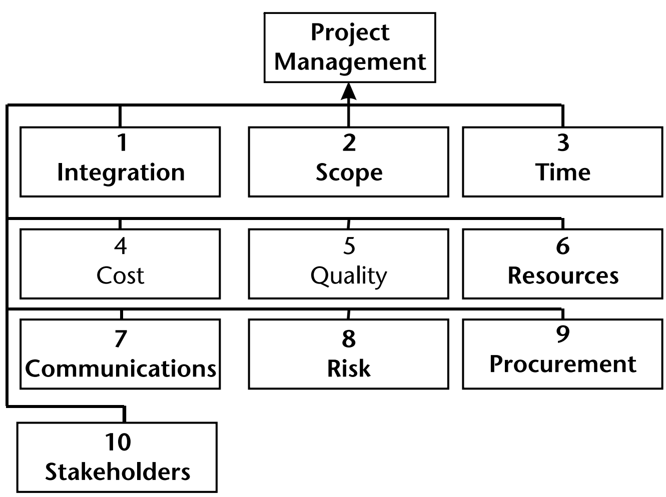


Figure 3.2 The Project Management Body of Knowledge Guide (PMBOK Guide) areas identify the project system.

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 areas, collected into five types of processes:

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

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

3.3.1 Project Integration Management

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

3.3.2 Project Scope Management

Project scope management includes the process leading to initiation of the project and scope planning, definition, verification, and change control. Primary outputs of the scope management processes include a project charter, the project Work Breakdown Structure (WBS) and detailed Statements of Work (SOW), 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 schedule. The planning and control processes defined by the PMBOK™ Guide do not include a way to handle decision branches in a project schedule. Assumptions define uncertainty sufficiently to permit defining a deterministic scope, cost, and schedule.

3.3.3 Project Time Management

Project time management includes defining the activities necessary to produce the project scope, sequence the activities, estimate activity duration, develop the project schedule, and control the project to the 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. The PMBOK Guide does not differentiate between common-cause variation and special-cause variation, but the latest revision to the Risk Practice Standard now does.

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

3.3.4 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 latest edition of PMI's risk management practice standard finally distinguishes between common-cause and special-cause variation, a substantial improvement. Although I never heard Dr. Goldratt use the distinction between common and special-cause variation and find that he and many of his followers lump them together under the title of variation, or "Murphy," I feel that dumbing down understanding uncertainty this way is a mistake. The distinction is critical to understanding buffers in CCPM.

3.3.5 Other PMBOK Guide Knowledge Areas

The other PMBOK knowledge areas, including quality, human resources, communications, procurement management, and, new to the fifth edition, stakeholder management, are all important, in varying degrees, to projects. They are important to any type of business. The scope of this text does not explore these areas simply as a method to focus.

3.3.6 Rolling-Wave Planning

Although the PMBOK Guide addresses alternative development cycles, including the spiral development cycle specifically addressed since the first edition of the PMBOK Guide, many organizations struggled 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 for the types of projects with which I worked. 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 3.5 describes this in a little more detail along with some other methods to treat the problem of lesser known requirements at the outset of a project.

3.4 Lean

Lean began with Taiichi Ohno's development of the Toyota Production System (TPS) starting in 1945. His principle objective was to "produce many models in small quantities" (Ohno, 1988, p. 2). By 1988, it had moved to "looking at the time line from the moment the customer gives us an order to the point when we collect the cash. And we are reducing that time line by removing non-value-added waste" (p. ix).

Ohno (1988) noted, "the greatest waste of all is excess inventory" (p. 54), "called the waste of overproduction" (p. 59). Project language usually does not address inventory, which I define for projects as "all of the work performed on projects that are not yet complete and producing their benefits." Project inventory directly affects project flow because new projects must wind their way through the pile of projects in progress and indirectly affects it by causing multitasking, a huge producer of waste. Shingo (1989) added, "process consists of four components: processing, inspection, transport, and delay operations. Of these, only processing adds value; the others can be viewed as waste" (p. 77). This perspective relates all of the waste types to project tasks as processes.

Womack, Jones, and Roos introduced the world to Lean thinking with *The Machine That Changed the World* (Womack, Jones, and Roos, 1990). They defined the principles of Lean production to include:

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

Rother's coaching and improvement katas (Rother, 2010) described the principles well. Womack and Jones (1996) expanded 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 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*. Others have added TOC to the blend calling it TLS (TOC, Lean, Six Sigma).

Lean is mostly defined by the focus to eliminate waste in the value stream. Dennis (2007) clarified three high-level types of waste: *mura*, unevenness or fluctuations in demanded work; *muri*, jobs that are hard to do; and *muda*, anything that does not contribute to value. Womak (2013) noted, "The inevitable result is that mura creates muri that undercuts previous efforts to eliminate muda." As we proceed, I will refer to two levels of reducing in Work in Progress (WIP). One provides a solution to mura and the second addresses thinking of too much WIP and multitasking as aspects of muri. I will only lightly address the "hard-to-do" aspect of muri in this book by recommending improvement processes such as Lean and Six Sigma, although reducing multitasking, a major emphasis of this book, also makes jobs easier.

Most Lean authors identify seven types of *muda* waste:

- Waiting;
- Unnecessary transport of materials;
- Overprocessing;
- Inventory;
- Unnecessary movement by employees;
- Defective parts;
- Overproduction.

Dennis suggested an eighth type of *muda* waste: knowledge disconnection. The disconnects can be within a company or between a company and its suppliers or customers. Lean processes seek to eliminate these forms of waste through the Lean tools such as value stream mapping and the continuous improvement.

All of the *muda* wastes apply to project work as well as they apply to production. Further, Womack and Jones also noted, "the general principle of doing one thing at a time and working on it continuously until completion" (Womack and Jones, 1996). I do not know why they did not include multitasking, the reverse of this principle, as one of the major forms of waste. It has turned out to be the most important form of waste for knowledge-work projects. CCPM Rule 1, Focus, directly addresses this principle.

The primary Lean approach addressed to reduce *mura* waste, the unevenness in production, is called *heijunka*, also called demand or production-leveling. Heijunka and Kanban combine to create low inventory and enable customer pull of product. In the TOC Drum-Buffer-Rope production model, the rope, which releases work into the system, provides a kind of *heijunka*. For CCPM you will see how Pipelining (Rule 3) performs this function.

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.

Dettmer (2000) provided an excellent comparison and contrasting 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 identified some significant advantages to synthesizing Lean and TOC, including the following lean tools:

- Poka-yoke (mistake-proofing operations);
- Statistical process control (SPC);
- Continuous improvement;
- Failure modes and effects analysis (FMEA), both product and process;
- Line stop;
- Cell design (meaning, in this case, establishing work centers around natural work groups);

- Team roles/responsibilities/rules;
- Graphic work instructions;
- Visual controls;
- Five "S" (The five "S's" are seiri, seiton, seiso, seiketsu, and shitsuke, which roughly translate into English as sifting, sorting, sweeping, standardize, and sustain, respectively. 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 identified the primary challenge of synthesizing Lean and TOC as relating 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." My approach to CCPM seeks to build on the strengths while avoiding the obstacles.

3.5 Agile or Light Project Management

The project management community has paid quite a bit of attention to light or Agile methods as a solution to the specific problems of projects involving information technology (IT). PMI recently added a version of the PMBOK Guide focused on Agile methods and now issues a certification specific to those methods. The processes originated from those methods were seen as bureaucratic, slow, demeaning, and contradicted the ways that software engineers actually work." Proponents sometimes characterize the Agile approaches as Lean project management. The symptoms of the problems leading to the Agile approaches were as described in Chapter 1: extensive cost and schedule overruns and failure to deliver error-free scope on most IT projects. Agile methods for IT projects include such methods as:

1. Rapid Application Development (RAD);
2. Joint Application Development (JAD);
3. Extreme programming;
4. Scrum;
5. Kanban.

Detailing the first four methods is beyond the scope of this text, but I will provide substantial detail on Kanban.

I do not agree with some of the claims that lead to proposing the Agile methods, such as "conventional project management does not work for IT projects." I have not heard people objectively skilled in professional project management (e.g., certified Project Management Professionals) make these claims. For example, Anderson (Anderson, 2003, p. 55) noted: "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…created the worst possible environment for managers…The result was heavyweight, traditional software methods." Anderson concluded, "The existing PMI/ISO model for project management is obsolete" (p. 60). I feel that such statements result from misapplication of the PMI methods.

Although Anderson later presented a masterful approach to deploying CCPM on IT projects, my own observations of IT organizations struggling with IT projects is that many do not understand and do not apply conventional project management well, if at all: quite the inverse of overapplication of "heavy" methods. Major shortcomings include poorly defined initial scope (e.g., lack of a WBS) and lacking or ineffective change control processes. Some of the methods posed as alternatives to heavy project management, characterized by the PMBOK Guide, are actually included in the PMBOK Guide processes, perhaps most notably the spiral development and rapid prototyping approach. While the Agile methods seem to provide effective approaches for leading small teams through small development efforts, I find them more to be supportive approaches to parts of larger projects rather than replacement approaches to comprehensive approaches such as the PMBOK Guide.

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 prescriptively followed in their entirety for all projects in all organizations. They represent menus from which to choose and to adapt to specific organizational needs. Although there is a natural human tendency to apply such standards prescriptively and thereby get bogged down in detail, that is not a problem with 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 IT projects fall into this category. Many other projects are not able to define all requirements at the outset. These include 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, use the rolling-wave planning approach. Create specific Project Plans for that which you can plan with the information that you have (including assumptions), and include in your plan the activities to update the plan 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.

Penultimately, all projects require an effective change control process to deal with the changes that will arise, including better definition of requirements. Change control comprises an essential part of Agile project management. Key sections of the PMBOK Guide address change control. IT 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 that does not effectively apply change control. Most admit to not applying change control, frequently because they did not define the project assumptions or scope initially well enough, or because they did not clarify the change control procedures with project stakeholders at the outset. Once they start applying change control, they find that they are able to be more Agile and achieve project success.

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

3.6 Kanban

I am describing Kanban in a separate section to draw attention to it and make it easy to find. Kanban was developed by Taiichi Ohno, the developer of the Toyota production system (Ohno, 1973). He is reported to have modeled it after observing U.S. grocery stores. The Kanban method is well established for just-in-time workflow for production systems. To my knowledge, David Anderson was the first to adapt it to knowledge work through his application in IT (Anderson, 2010). When I read Anderson's work, I realized it was not limited to IT. It is not even limited to knowledge work; his approach to Kanban applies to all types of work.

I found Kanban as a solution to two major issues that I had been experiencing with CCPM implementations. The first issue was reducing multitasking at the individual level. Although organization leaders agreed with the waste caused by multitasking, they were having much difficulty with reducing it, particularly at the outset when introducing CCPM to an organization steeped in multitasking (that is, nearly all modern organizations). The second major issue was the conflict caused by nonproject work. Many organizations use the same resources for work on projects and to perform other work such as responding to field problems or proposals. They had no way for first-line supervisors to make priority decisions on which tasks their people should work, so they caused the workers to multitask between project work and nonproject work. Anderson's approach to Kanban, which I now call project Kanban, provides a tool to resolve both issues.

The Japanese word Kanban means a posterboard or sign. In production processes it usually is a bar code, but in knowledge work you can best think of it as a Post-it® note with information about a task. It can be a project task from a project schedule or a task of nonproject work. The key point in project Kanban is that it is a task that can be performed by one supervisor's work team. Figure 3.3 illustrates an example Kanban card for project or nonproject work.

The project Kanban process prioritizes all work performed by a work team to limit the WIP. Limiting WIP (also known as just-in-time) greatly reduces the cycle time of both projects and nonproject work. When it is done by eliminating multitasking, it also greatly reduces quality defects.

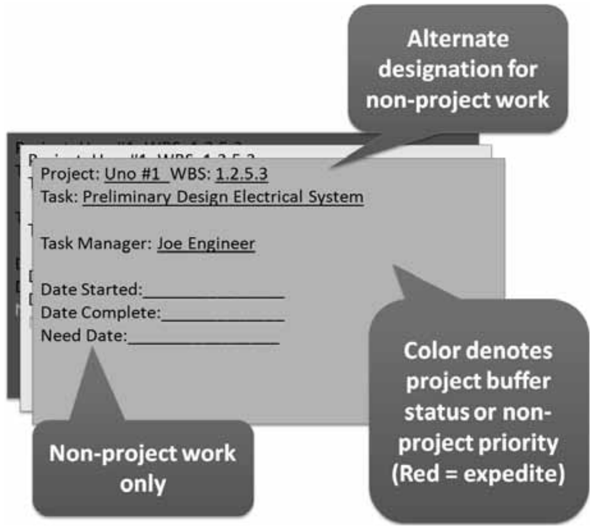


Figure 3.3 Example Kanban card. (Yours may vary to improve readability at a distance.)

Kanban provides a visual control tool that the project team and anyone who walks through the work area can see at any time. Thus, it randomly reinforces the idea of focus on one task at a time to limit WIP. The visual control is a Kanban board displaying the Kanban cards. It can be in various forms, but the most common form to start is a group of Post-it® notes on sheets of flip-chart paper taped to a wall where everyone can see it. The workflow and WIP limits are inscribed on the background. Other types of boards are used frequently and now many teams use an electronic display on a large-screen TV. A caution: the board must be visible in the work area at all times; it is not enough to just use an electronic implementation. The reason is that the visual control provides random reinforcement of WIP control any time someone sees it. That does not happen when it is just available on computer screens. Figure 3.4 illustrates an example Kanban board display.

Figure 3.4 illustrates the key features of a Kanban board. Although layouts differ, these key elements are essential:

1. An input queue for the available tasks to be placed on the board.
2. Columns for each work process step. The minimum case is three columns: input queue, working, and done.
3. WIP limits for each column.
4. Identification of the performer for active tasks.
5. Identifiers for tasks that are stuck or have issues.
6. A way to indicate priority tasks (e.g., color or type of card).

To emphasize, visual control provides multiple benefits. It:

- Simplifies workflow management (Rule 3);
- Clarifies work process;
- Enhances teamwork;
- Reinforces WIP control (Rule 3):

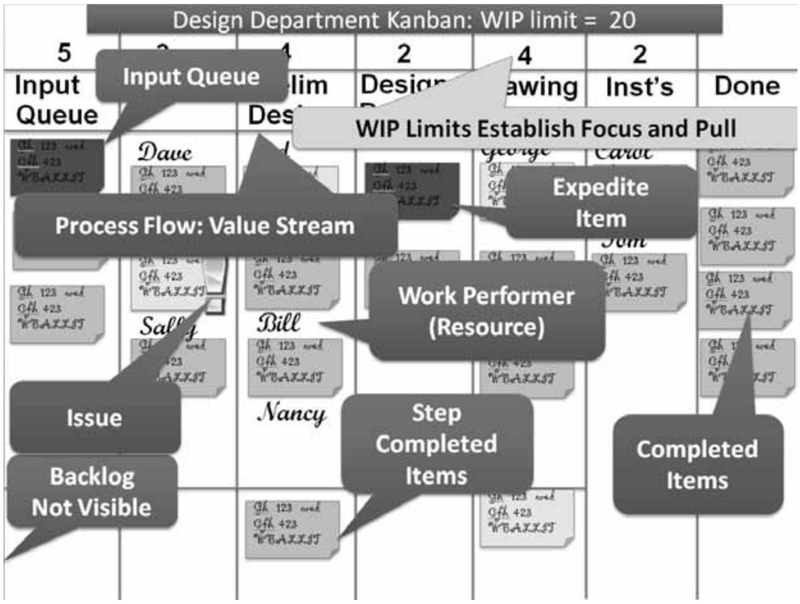


Figure 3.4 Example of a Kanban board display.

- Makes work easier;
- Focuses on the right tasks;
- Performs tasks the right way (i.e., with focused attention: Rule 1).
- Reinforces management gemba (the production floor) walks;
- Enables transparent decision-making.

Kanban requires certain disciplines by the team. The team needs a method to promote candidate tasks to their board for execution. This usually happens primarily in a weekly meeting by the team leader and the team's customer representatives. The team discipline is to only work on tasks that have been promoted to the input queue on the board and to control the number of active tasks on the board to the preestablished WIP limits (number of cards). The weekly meeting may need to be supplemented by ad hoc revisions to the input queue during the week for emergency work or changing priorities. The process for making those priority changes needs to be understood and accepted by all the customers of the team.

Kanban teams normally use a brief stand-up meeting in front of the board every day to highlight any issues for the team leader to get resolved. They can also use that time to move any tasks on the board that they have not moved during the course of the day or to highlight tasks they have moved so that others know that they are available to be pulled and worked on.

A major feature of project Kanban is that work teams can start it immediately and independently of other work teams. Thus, while an organization is struggling to learn how to create CCPM schedules and limit work demand at the project level (also known as Pipelining), work teams can start using team Kanban anyway and enjoy the benefits of local WIP control. All they need to be able to do is identify discrete work tasks that the team must perform.

Ohno (1974) described six rules of Kanban. They apply to project Kanban as well. In the following list, I also note how the new project execution CCPM rules relate to some of the Kanban rules:

1. The later process picks up work from the earlier process (pull: perform tasks to priority not dates, CCPM rule 1: Focus).
2. The earlier process produces only what is needed by the later process (avoid overproduction: CCPM Rule 3: Pipeline).
3. No items are made without Kanban.
4. All items carry a Kanban.
5. The defective product is not passed on.
6. Reducing number of active Kanban improves process (reducing WIP).

As suggested above, the pulling of tasks is by team members into active work columns. A team member who completes a task is free to pull the next task from the input queue that he or she is qualified to perform.

3.7 Quality Focused Improvement

Developed by Motorola but made famous by General Electric, Six Sigma adds to the approaches of Total Quality Management (TQM). The Malcolm Baldrige National Quality 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 National Quality Award (NIST, 2002) 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 and drives process improvement and cost savings.

ISO 9001:2000 Registration is a product/service conformity model for guaranteeing equity in the marketplace and concentrates on fixing quality system defects and product/service nonconformities.

The Baldrige Criteria for Performance Excellence focuses on performance excellence for the entire organization in an overall management framework and identifies and tracks 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. One Six Sigma book asserts that it solves all of the problems that TQM experiences (Pande, Neuman, and Cavanagh, 2000, pp. 43–49). I consider TQM to still be quite successful, when applied appropriately, and Six Sigma to be part of the ongoing process of improving TQM processes.

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. Hendricks and Singhal (1999) published results demonstrating performance measures for TQM award-winning firms outstripped 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 was performing well again in 2004. The trend continues: Witjaksono (2012) concluded, “The findings of this study indicate that: 1) The level of TQM practices in the TQM firms is higher than non TQM firms; 2) The organizational performance with the level of TQM practices above the average have better organizational performance than companies with TQM practices below average. Based on the findings of the study, it can be concluded that TQM practice was proven as a powerful approach for the organization which desired to accomplish excellent performance.” I view Six Sigma as a continuous improvement to TQM.

Six Sigma takes its name from a long-term goal of seeking to reduce defects such that the process output conforms with customer requirements within plus or minus 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 plus or minus 1.5 sigma). Sigma is the statistical measure of the process standard deviation. Six Sigma considers variation as the evil on which to focus.

Six Sigma builds on Dr. 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. I will focus on these points with CCPM. Although TOC also applies understanding of variation to develop and deploy simple solutions, it avoids Six Sigma’s rigorous application of statistical tools. All of the Six Sigma approaches can complement applying CCPM if you avoid the danger of suboptimizing on a single process and instead focus on exploiting the system constraint.

3.8 System of Profound Knowledge

Dr. W. Edwards Deming, the man that most people consider the father of Total Quality Management (TQM), never defined TQM. Deming described his approach in seminars and books (1982, 1993) and, although a great advocate of operational definitions, chose to never offer one for TQM. Instead, he preferred to discuss the matter in terms of his 14 points, or “Principles for the Transformation of Western Management.” He supplemented these points with identified diseases and obstacles to achieving the transformation that he preached.

In later life, Deming brought together the overall methods in which he believed under the title of “A System of Profound Knowledge.” Deming (1993) defined a system of profound knowledge 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 the parts interconnected. He identified four segments for discussion, but emphasized that they cannot be separated. The four elements are:

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

Figure 3.5 illustrates the interrelationship of the four elements. The following relates the elements to the project management system.

3.8.1 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 system 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 healthcare 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 cost. 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

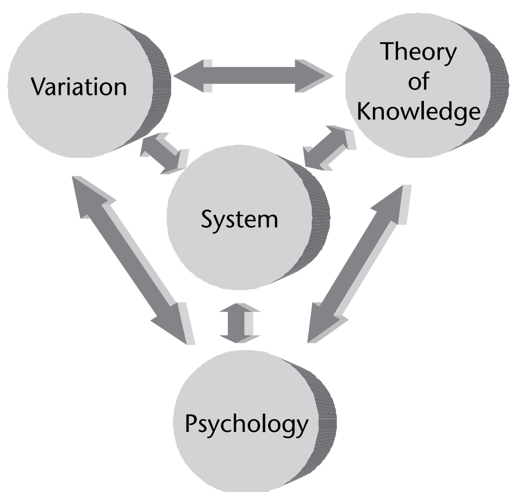


Figure 3.5 The four areas of profound knowledge interrelate.

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 an immense impact on the performance of the system.

Dr. Deming drew a sketch, similar to Figure 3.6, on a blackboard in Japan in 1950. He attributes the subsequent success of postwar Japan in large part to the understanding conveyed by this figure. His description of the system starts with 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 towards 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 Project 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 makes reference to the following material.

System Dynamics

Senge (1990) 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 a project system) work. The following list gives the laws and one instance of how each law applies to the project management system.

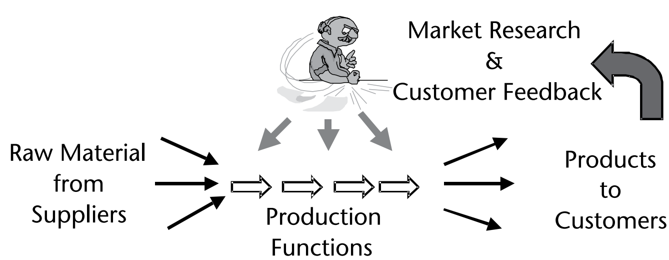


Figure 3.6 Dr. Deming's sketch of a business system emphasized interrelationships and feedback.

1. Today's problems come from yesterday's solutions. Management was unsatisfied with an overlong 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 on which they can work. Working on multiple tasks means that all tasks take longer, as people can really only work on one task at a time. The others are doing nothing while one task is worked on. Projects become longer, and efficiency decreases.
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" (Brooks, 1990) 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 including more detailed schedules. This often helps on a project performed 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 and schedules 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 really needed two more days of testing 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 Cape Kennedy in Florida. The cause is a seal design made and tested in Utah years before but not previously subjected to 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 that 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.
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 at once. The Critical Chain multiproject process completes more projects much faster; individual project duration decreases and the number of projects complete in any time interval increases. However, you must delay the start of projects to get the benefit.
10. Dividing an elephant in half does not produce two small elephants. Senge relayed 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 of deterministic task schedules implicit to printing out start and finish dates for thousands of tasks.

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 small interest rate can lead to 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 noted 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 were 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 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 that may leverage improvements in the project system.

Unintended Consequences

The linkage and correlation between parts of a system mean that changing any part of the system may influence other parts. As noted in the laws of the fifth discipline, the change may be in a desired direction or not, it may be large or small, and it most likely will not be in the same time and place that caused the change. Many people, especially those prone to fiddling with social systems, talk in terms of unintended consequences. Hardin (1985) made the point that, from the ecological view of systems, there is no such thing as unintended consequences. When you change part of a system, other parts change. That is it. You can count on some of those changes being undesirable to one or more perspectives of the system. 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 contingency. We may also cause 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.

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 versus cooperation between departments often causes such destruction. Senge (1990) and Deming (1993) illustrated numerous examples of how government attempts to improve things often lead to destruction of the very system they were hoping 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 latter 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 noted: "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."

3.8.2 Understanding Variation and Uncertainty

I returned, and saw that under the sun, that the race in 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.

Understanding variation is essential to making any real system operate. Popper (1979) in an essay titled, "Of Clouds and Clocks," described 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 at the human scale. Popper wrote, "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 means indefinite, indeterminate, and not certain to occur, problematical, not known beyond doubt, and/or not constant. All predictions 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 on 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 including our knowledge about the result, a measure of the predictability of the variation. For example, the results of any task in a project will vary if you did 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 who do the task next time may not be the same ones who did it last time. 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, and how well we measure conditions at one point in 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. However, they can predict seasonal trends reasonably well.

Starting in the seventeenth century, mathematicians and scientists have sought to improve the ability to predict the world further over into the cloudy region. At the same time, science kept moving the cloudy region to include more 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 study of complex adaptive systems.

Common and Special-Cause Variation

Probability and statistics are science's weapons of choice to deal with cloudy systems. Shewhart (1986), a mentor to Dr. 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 theorem 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. 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 repeatedly 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:

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

Dr. 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 automatically adjust the tool location based on each measurement to 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.

I experienced a similar mistake. A presenter was explaining how one might use (or misuse, as it turned out) statistical control charts derived from an electronic Kanban system and suggested the team should look for a cause and make improvement for each measurement point below the mean on the control chart. That approach is a sure method to increase future 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 schedule are uncertain. Performing each of the tasks within a project schedule is a single trial of a system (the project task performance system), and therefore it is impossible to predict 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. Until the issue of the latest *Practice Standard for Project Risk Management* (PMI, 2009), The PMBOK Guide and the supporting literature that I have examined failed to differentiate between common-cause variation and special-cause variation. This remains a major oversight in most of the current theory.

3.8.3 Psychology

Several properties of the human mind lead to individual behavior that seems to resist change. B.F. Skinner (1953) described 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 gives positive reinforcement and learn to avoid doing things that do not lead to positive reinforcement or that help us avoid negative reinforcement. Positive reinforcement is something you like. Negative reinforcement is something you do not like. Positive and negative reinforcements vary from individual to individual. Skinner noted, "A reinforcing connection need not be obvious to the individual reinforced."

Figure 3.7 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 the five senses or more removed methods 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 be somehow stored within the brain. As it defines a (perhaps rudimentary) model of the world (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 3.7 illustrates that beliefs impact to what you pay attention, how you interpret what you sense (perception), what your motivations (needs) are, and the decisions you make on how to act in the world so as to increase your rewards and decrease your negative reinforcers. This influence is mostly unconscious. In other words, you see it because you believe it.

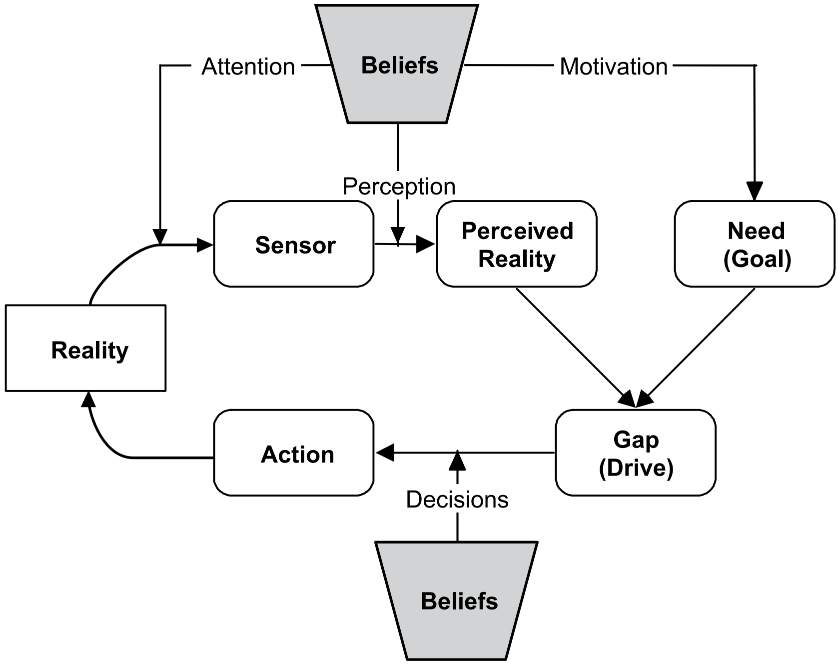


Figure 3.7 Control system view of human actions (behavior).

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 over-simplified 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 the reward/punishment concepts; even though Skinner identified and described in depth and proved by experiment, that punishment does not work.

Worse yet, research with humans conclusively demonstrates, 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. Kohn (1993) described 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 lose intrinsic motivation.

This is not new news, but much of modern management does not get it. Frederick Herzberg (1966) 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 Dr. Deming demonstrated with his bead experiment (Deming, 1993), 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 that tune in three notes. It is 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 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. Because 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 position. Furthermore, Skinner demonstrates that extinguishing behavior established by operant conditioning can take a long time. The organism will continue to emit 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 us the most, when offering judgments about probability. Thus, for example, you will often hear statements such as, "All scientists (programmers, engineers, and so forth) tend to underestimate how long it will take to do a task." When pressed for data, people offer anecdotal examples but no actual data. When data exists, 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. I will dig into the implications of this later. 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. Because the present system was not designed with this connection in the forefront, you may expect to find some problems in this area. Section 4.3 demonstrates that the core conflict leading to most of the observed undesired effects with the current project system stem from a mismatch between individual psychology and the project system goal.

3.8.4 Theory of Knowledge

Popper (1979), in an essay titled Conjectural Knowledge stated, "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, is far from 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 went 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 suggested 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 3.8 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 the scientist proposed the new theory. 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

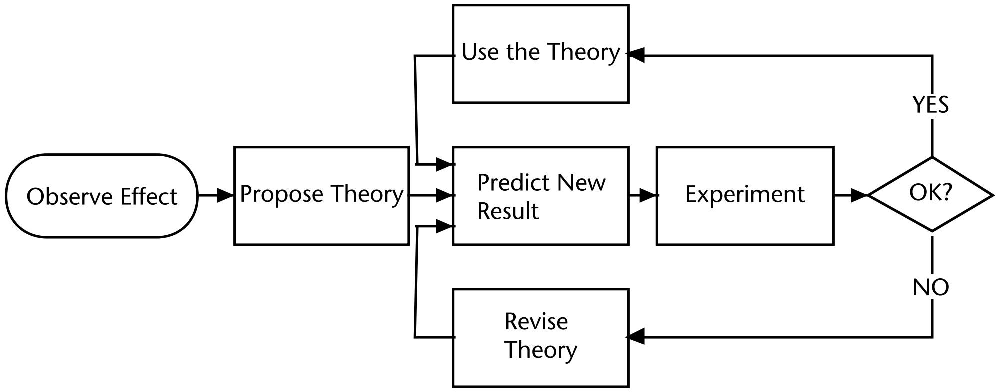


Figure 3.8 The scientific method checks the validity of a theory by experiment. No theory is ever proven. It is accepted as good enough to use until rejected by a single experiment or replaced by a theory that better predicts reality.

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.

A commonly used example of the scientific method is Newton's laws of motion and gravitation. Before Newton, mankind gathered much data on the positions of the Sun and planets. They developed correlations to quite 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.

Newton's laws worked better than his predecessors did because they extended beyond observations. Newton's laws allowed prediction beyond the realm of the observed and allow us to put men on the moon and send spacecraft to Jupiter. This is impossible using 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. This 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 other theory of project management that 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.

3.9 Theory of Constraints

The basic Theory of Constraints (TOC) is a commonsense way to understand a system. According to TOC, "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 go to zero. Therefore, a constraint limits any system with a nonzero output. Figure 3.9 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* (Goldratt, 1990), Dr. Goldratt stated: "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." Dr. W. Edwards Deming noted in *The New Economics* (1993), "We learned that optimization is a process of orchestrating the efforts of all components toward achievement of the stated aim."

A physical chain provides the most commonly used prop to describe TOC. 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 (see Figure 3.10).

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 schedule can have only one longest path. The only case that would have this not true is if two more paths are exactly the same lengths. 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. William Dettmer posed the following list in his book, *Goldratt's The Theory of Constraints, A Systems Approach to Continuous Improvement* (Dettmer, 1995):

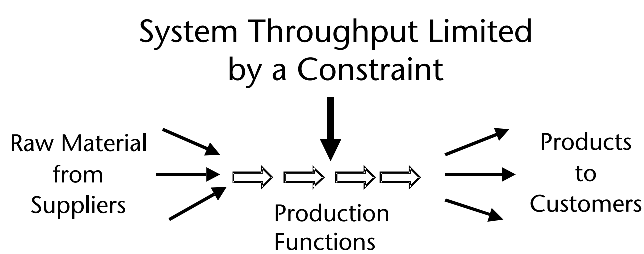


Figure 3.9 TOC limits the output of a system by a constraint.

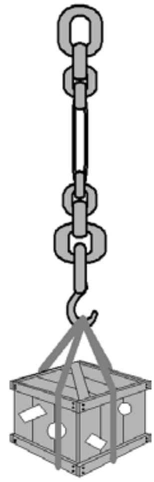


Figure 3.10 A physical chain illustrates TOC in action: the weakest link constrains the strength of the chain.

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.

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 undesirable effects (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 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. Physical constraints are relatively easy to identify and simple to eliminate. Policy 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 this list. The core problem is a problem that, if removed, would begin to cause the system to change undesired effects into desired effects. In other fields, it is called the root cause. He 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 undesired effects 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 was the earlier statement of the present understanding.

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

Unfortunately, despite my initial enthusiasm for the thinking process, my observation is that it has achieved little in the way of practical results and is not widely practiced. I have only seen it presented in TOC conferences and literature and in most of those cases it is presented without evidence of practical results. It also appears to me to be presented as an explanation for a solution conceived otherwise rather than a data-driven bottom-up analysis.

3.9.1 The Throughput World

Dr. Goldratt found that, most of the time, system constraints trace back to a flawed policy rather than to a physical constraint. In *The Goal* (Goldratt, 1984), 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 last century (twice the history of project management systems) and have changed little since. When they were developed, they were based on assumptions (no longer listed) about the design of business enterprises.

Dr. 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.

Dr. Goldratt defined a new way of accounting, called the throughput world. It rests on three definitions:

- *Throughput (T):* All of the money you make from selling your product. (Revenue minus raw material cost.)

- *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 inventory are treated the same.)
- *Operating expense (OE):* All of the money you spend to produce the throughput that does not vary with the quantity of throughput (e.g., management, buildings, energy, marketing, and salaries).

While some major accounting authorities around the world have endorsed this method mainstream adoption proceeds slowly.

The cost world was not so bad when it was developed around the turn of the last 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 was 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 from those dictated by the cost world.

For example, in the cost world, managers measure 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 make (raw materials) and to store. So it hurts cash flow and reduces disposable cash at the plant. Our accounting system counts inventory as a good thing (an asset), but it is 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.

In contrast, what do people normally consider their biggest competitive edge in knowledge industries? People. What are people on 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 meaning of the dilemma facing managers is to apply one of the thinking process tools invented by Dr. Goldratt: the evaporating cloud. Figure 3.11 illustrates the throughput world, cost world evaporating cloud. Block A represents a common objective that 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.

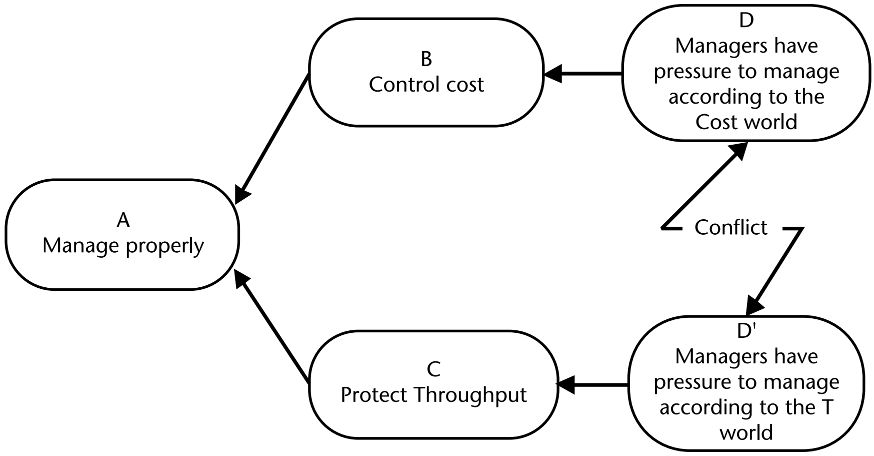


Figure 3.11 The throughput world/cost world evaporating cloud exposes the manager's dilemma.

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 an immediate conclusion that T is the most important variable. Improvements in throughput 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 focus on OE. You can reduce OE in any part of the system, and the sum of the OE reductions adds 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 has not everyone adopted throughput accounting and thinking? Inertia. Chapter 4 explains the importance of thinking and operating in the Throughput world relative to projects

3.9.2 The Production Solution

Dr. 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 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 Dr. Deming's meaning of the aim of a system.

Dr. Goldratt's books, most notably his initial international bestseller *The Goal*, demonstrated how he invented and used the Theory of Constraints 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 variable cost (e.g., raw material cost). 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 to never starve the constraint and also to not overload it. Overloading the constraint (that is, producing more than it can process) creates excess in-process inventory (piles of incomplete work in front of the constraint). The Rope transmits information from the Drum to the release of work in order to never overload or starve the constraint and to limit the buildup of WIP inventory. It enables a form of production-leveling or *heijunka*.

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 3.10 illustrates a production system. Compare it to Figure 3.6, and note that this represents the inner workings of the overall business system depicted by Dr. Deming. Production is a subsystem of the overall business system, just as the circulatory system is a subsystem of the human body.

Although Dr. Goldratt used the background of a factory that produces hardware products in *The Goal*, the general nature of TOC 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 nonprofit organization. The systems include government. Nonprofit and government systems obviously have a different goal (aim) from for-profit business.

Figures 3.6 and 3.12 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 process has some amount of variation, often referred to as statistical fluctuation. Because workstations downstream of other workstations need the parts from the upstream workstations, they are dependent on the upstream workstation. 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 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 protective 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

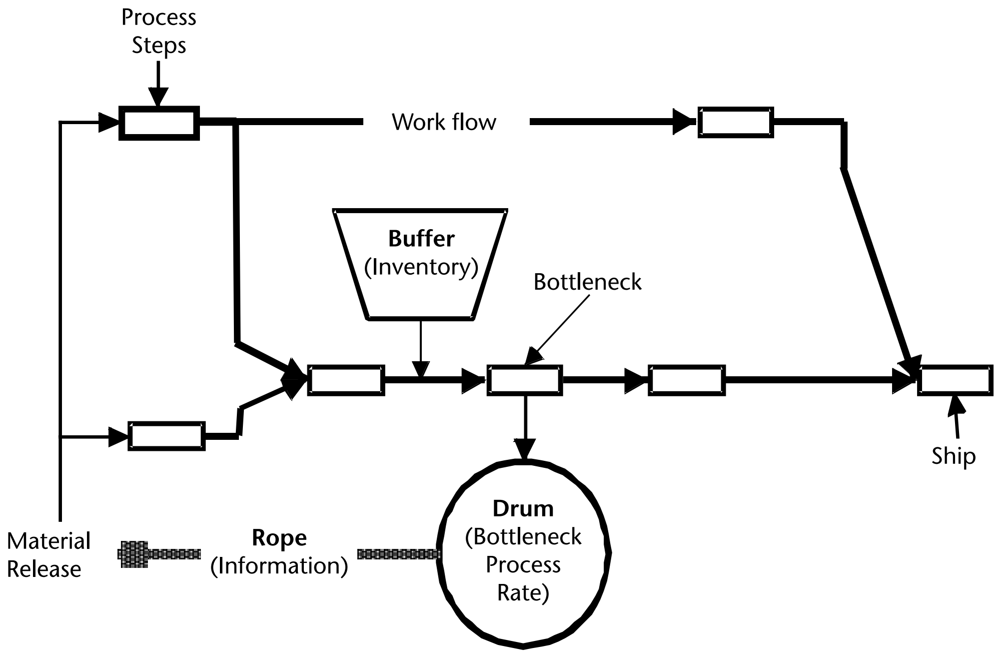


Figure 3.12 Drum-Buffer-Rope is the solution to operating a production facility using the TOC. This solution operates to the global optimum (the system goal), and accounts for the combination of statistical fluctuations and dependent events.

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 to have reserve capacity to make up for fluctuations.

Queuing addresses the buildup of lines waiting to be processed. Because there is variation in both the arrival time of things on which to be worked and in the processing times, it is a probabilistic process. It is well defined mathematically. One way of describing it considers the probability distribution of the arrival rates and of the processing rates.

I ask my students to consider a buildup of a line with which they are familiar, for example, in a bank, supermarket, or for passport processing or anything else. I ask them to consider a case in which the average arrival rate exactly equals the average processing rate, for example, each at one per minute. How long is the average line? Consider this yourself.

Although I occasionally get one or two who know queuing theory (and ask them to remain silent), after a few minutes, most venture answers of no line or at most one person. Figure 3.13 illustrates the reality of this situation: the line grows to be infinitely long when the average arrival rate approaches the average processing rate. Of course, it would take infinite time for the line to become infinitely long and as the line gets long, new arrivals balk (i.e., leave the line). This process, although well known, simply is not intuitive. Note that if you plan to load resources above about 75%, you guarantee that most of the time something spends

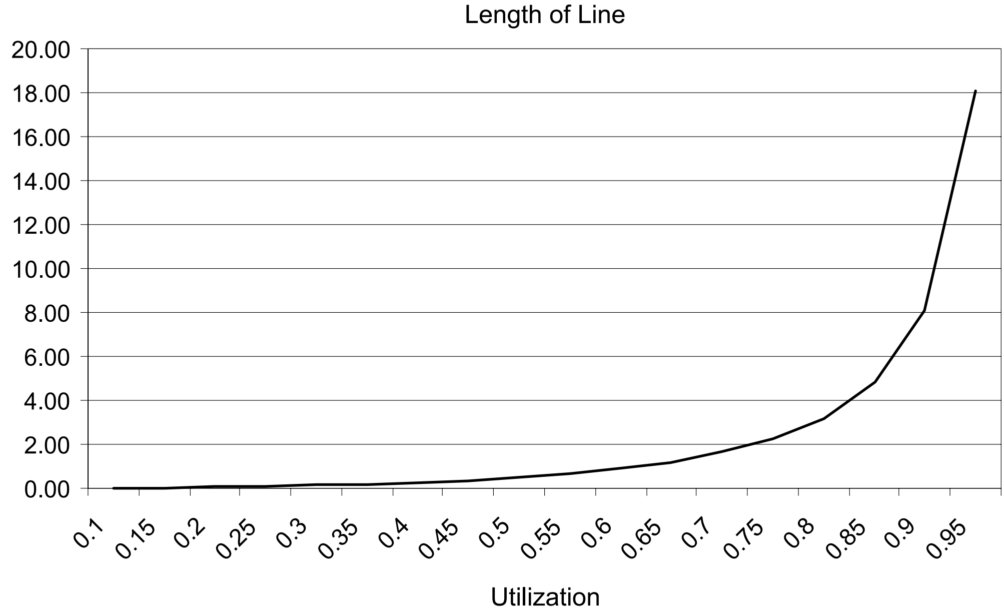


Figure 3.13 The queuing curve plots the length of line or average wait time versus the system utilization.

is queuing time instead of processing time. This works as well for project resources as it does for resources in a production line.

This understanding is a major reason that the TOC is able to make such an immediate impact once people understand it. Managers design and operate most current systems without the 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 to short-term throughput. Once they understand the theory, identify the constraint, and improve its throughput, the system throughput increases immediately.

The computer systems that Dr. Goldratt was selling before he invented the TOC and all other factory control systems failed to account for the impact of the system constraint combined with these statistical fluctuations and workstation dependency. Because the actual fluctuations are statistical, they are unpredictable. You can only predict the general behavior over a period of time and 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 schedule did not cause the system to make more money and that adding more detail to Project Plans does not make projects more successful.

In *Critical Chain*, Dr. 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 exist 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: late delivery, 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. (Actually, in production the physical material buffers also relate to time. A pile of a certain size provides a certain time 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 activities for work based on the input from buffer reporting (using prioritized task lists for Rule 1: Focus);
- The decisions made in buffer management on when to intervene in the process (Rule 2: Buffer);
- Release of projects to the system based on the ability of the constraint resource to process the project tasks (Rule 3: Pipelining).

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, a nonprofit organization, or a government agency. A middle manager of a former client stated, "Work in [their 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 on which people have tried it, 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* (Goldratt, 1994) shows how the 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* (Goldratt, 1984) demonstrated 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). 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 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 you get more operators? If you have extra operators (so the longer calls and variations in when calls arrive can be handled), this means that 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 you are contributing to the company goal? Do you know where the company constraint is and how human resources might influence it? Dr. Goldratt defines several necessary conditions for achieving the goal of a company. One of these is, "Satisfied and motivated 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 impacts operating expense in several ways, including their own contribution (cost), and impacts they may have on company salaries and benefits through salary and benefit policies and union agreements.

3.9.3 Five Focusing Steps

Having realized the goal of the system and the fact of a constraint, Dr. Goldratt invented the five focusing steps as a process to get the most out of a system in terms of the system goal. Figure 3.14 summarizes these steps.

Identify the System's Constraints

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 a weakest link of 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. Because 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, the 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 cause of the reasons that the system is not performing better. It is the root cause of one or more undesirable effects in the system. To eliminate these undesirable effects, you have to first identify the core conflict.

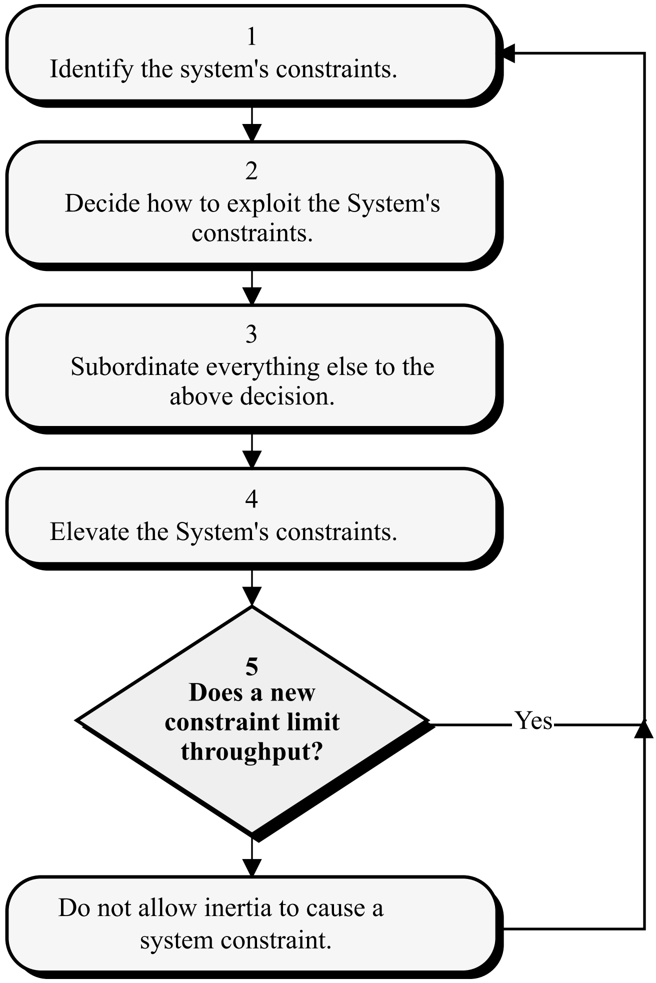


Figure 3.14 The five focusing steps represent the TOC approach to ongoing improvement.

Decide How to Exploit the System's Constraints

Exploiting the system constraint is getting the most out of the weakest link of the chain: utilizing the constraint to the fullest for organization throughput. There are usually many ways to do this. For example, in a production facility, one way to improve throughput of the production system is to change the way that 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 our changes will achieve the desired effects that you want to have.

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

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 many 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 caused people to build parts for products that were not going to sell immediately, causing cash outlay for parts, and possibly plugging the system constraint, again impacting products that customers wanted and would lead to immediate income.

As project management has been in existence for over 40 years with little change, is it not likely that there are some assumptions, policies, or artificial constraints that do not work well anymore? 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.

Elevate the System's Constraints

While exploit and subordinate provide the means to make better use of what you have, elevate means to increase the quantity of what you have: hire more resources or buy more equipment. Elevate comes after exploit and subordinate because elevate always costs money and takes time. Exploit and subordinate usually are free and immediate.

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

As you continue to exploit, subordinate to, or elevate the current constraint you always eventually unearth another constraint. It may be lurking a few capacity percentages above the current constraint, or you may be able to improve the system many tens of percentages before you uncover the next real constraint. This is not a problem. Instead, it 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 to not let management's inertia become the system's constraint.

3.9.4 The Thinking Process

Figure 3.15 illustrates the overall thinking process flow and identifies the primary tools. Dr. Goldratt designed the thinking process to answer three questions:

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

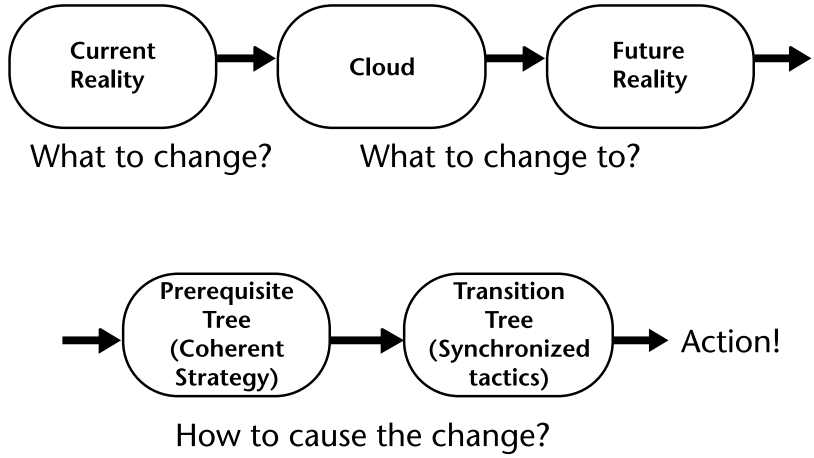


Figure 3.15 The thinking process leads us from undesired effects, through the core problem, to successful implementation.

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

Dr. Goldratt developed the tools necessary to apply the thinking process. Figure 3.15 illustrates the process. It starts with a current reality tree (CRT) to define the system causing undesired effects (UDEs). It builds to a future reality tree (FRT) with the UDEs replaced by their counterpart desired effects (DEs). It ends with a synchronized plan to move the system from the CRT to the FRT. In addition to their use in the thinking process, the tools (other than the CRT and FRT) have stand-alone application. Next I describe the tools but will not use most of them 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 uses the evaporating cloud, which I believe to be the most elegant stand-alone TOC tool.

Most people find the list of TOC thinking process tools and associated acronyms intimidating. 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. Noreen, Smith, and Mackey (1995) reported that, even after this training, only a limited number of people are able to create significant solutions. Their book is becoming somewhat dated. The process, tools, and training have changed since their survey, in some respects to make them easier. I am not aware of more recent survey data, but my inquiries suggest that the Noreen, Smith, and Mackey conclusions still apply: the process is little used after training and, when used frequently, is not successful other than to explain history.

You do not need to understand all of the TOC tools to successfully apply CCPM. The reason for mentioning 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. For additional knowledge on the thinking process, please see Dettmer (2010).

3.10 Change Management

Change management is how to cause the change. 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 quite effective in understanding what is going on, and how to cause the necessary changes in behavior.

Braksick (2000) provided a research based approach effective in the business environment. Her approach, based on the ABC model (Antecedent→Behavior→ Consequence) of operant conditioning, leads to a method with similarities to the Six Sigma approach to process analysis and improvement. She described PIC/NIC analysis as a way to lead behavior change. PIC/NIC analysis seeks to determine consequences that reinforce or extinguish behaviors by being positive or negative, immediate or future, and certain or uncertain.

Daniels and Daniels (2006) provided a thorough basis and methods for using the vast quantity of psychological research on behavior. For some unknown reason, they claimed several times that PIC/NIC analysis is not scientific, but they nonetheless described in detail how to use it (Chapter 6). They also provided a vast amount of practical guidance on behavior change.

Kotter (1998) described 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 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.

Dr. Goldratt proposed a layers-of-resistance model, referred to 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 the Kotter model at the organizational level, supplemented by the Braksick approach and the Daniels and Daniels approach at the individual level, produce a more effective action-oriented approach. Chapter 11 describes an effective implementation approach using these models.

3.11 Synthesis

Although each of the management theories devolves into considerable detailed complexity, two points of view 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 company could have very little success with the same set of tools simply by one company focusing on the constraint and the other not focusing on the constraint. 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 suspect that the successful ones may 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 Pande, Neuman, and Cavanaugh's Chapter 3 (2010), "Why Is Six Sigma Succeeding Where Total Quality Failed?" Both succeed when applied to the constraint. 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. This effect is not unique to TOC and I will cover approaches to remedy it in Chapter 11.

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

3.12 Summary

This chapter showed how thinking from four related management disciplines combines to improve the generic system for project management. There is considerable overlap among these disciplines and little disagreement on fundamental values and principles. I hope you agree from 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 understand how to best apply TOC to the project management domain.
- Lean, Six Sigma, and TOC all operate with Dr. W. Edwards Deming's points of profound knowledge: appreciation for a system, understanding of variation, a theory of knowledge, and understanding of psychology.
- TOC provides a logical process to improve a system, answering the questions: What to change, what to change to, and how do we 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 undesired effects 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.
- Project Kanban provides multiple benefits:
  - Supports workflow (pull);
  - Integrates project and nonproject work;
  - Establishes priority for all work;
  - Limits WIP to reduce the waste of multitasking;
  - Reduces cycle time;
  - Reduces quality defects;
  - Increases throughput;
  - Levels demand;
  - Relies on local control to enhance agility;
  - Enables key support processes;
  - Supports continuous improvement.
- 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 2 and the theory background in this chapter set the stage to develop an improved theory for project planning and execution. TOC provides one tool set and it provides a strategy to apply the tools of Lean and Six Sigma for that purpose.

References

Anderson, David J. (2003). *Agile Management for Software Engineering*. Upper Saddle River, NJ: Prentice Hall.

Braksick, L. (2000). *Unlock Behavior, Unleash Profits*. New York: McGraw-Hill.

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

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

Deming, W. Edwards. (1982). *Out of the Crisis*. Cambridge, MA: MIT Press.

Deming, W. Edwards. (1993). *The New Economics*. Cambridge, MA: MIT Press.

Dennis, P. (2007). *Lean Production Simplified*. New York: Productivity Press.

Dettmer, H. William. (1995). *Goldratt's Theory of Constraints: A Systems Approach to Continuous Improvement*, University Bookstore (now available through ASQC Press).

Daniels, A., and Daniels, J. (2006). *Performance Management*. Atlanta, GA: Performance Management Publications.

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

Dettmer, H. W. (2010). *The Logical Thinking Process: A Systems Approach to Complex Problem Solving*. Milwaukee, WI: ASQ Press.

Goldratt, Eliyahu M. (1984). *The Goal*. Great Barrington, MA: North River Press.

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

Goldratt, Eliyahu M. (1990). *Theory of Constraints*. Croton-on-Hudson, NY: North River Press.

Hardin, Garrett. (1985). *Filters Against Folly*. New York: Viking.

Hendricks, Kevin B., and Singhal, Vinod R., (1999). *Don't Count TQM Out…Evidence Shows Implementation Pays Off in a Big Way*. Quality Progress.

Herzberg, Frederick. (1966). *Work and the Nature of Man*. Cleveland, OH: World Publishing.

Kotter, J. (1998). *Leading Change: Why Transformation Efforts Fail*. Harvard Business Review on Change. (pp. 1–20). Boston, MA: Harvard Business Review Publishing.

Kohn, Alfie. (1993). *Punished by Rewards*. Boston, MA: Houghton Mifflin.

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

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

Ohno, T. (1988). *Toyota Production System: Beyond Large-Scale Production*. Portland, OR: Productivity Press.

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

Popper, Karl R. (1979). *Objective Knowledge: An Evolutionary Approach*. Oxford, U.K.: Clarendon Press.

PMI. (2003). *Organizational Project Management Maturity Model Knowledge Foundation*. Newton Square, PA: Project Management Institute.

PMI. (2013). *A Guide to the Project Management Body of Knowledge*. Newton Square, PA: Project Management Institute

PMI. (2009). *Practice Standard for Project Risk Management*. Newton Square, PA: Project Management Institute.

Rother, M. (2010). *Toyota Kata*. New York: McGraw-Hill.

Senge, Peter. (1990). *The Fifth Discipline*. New York: Doubleday.

Shewhart, Walter A. (1986). *Statistical Method*. New York: Dover.

Shingo, S. (1989). *A Study of the Toyota Production System*. Portland, OR: Productivity Press.

Skinner, B.F. (1953). *Science and Human Behavior*, London, U.K.: The Free Press, Collier Macmillan.

Wikipedia. (2004). *Agile,* http://en.wikipedia.org/wiki/Agile_Methods. Accessed June 21, 2004.

Witjakson, A. (2012). *The Differences of TQM Practice and Organization Performance Between TQM Firms and Non TQM Firms*, 2nd International Conference on Management. Economics and Social Sciences (ICMESS 2012) June 30–July 1, 2012, Bali.

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

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

Womack, J. (2013). http://www.lean.org/womack/DisplayObject.cfm?o=743. Accessed September 15, 2013.

CHAPTER 4

The Direction of the Solution

4.1 Deciding What to Change

The most important decision you make when you go about improving anything is 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 the system worse by making a new constraint more restrictive than the old constraint, but you cannot make the system substantially better by improving a nonconstraint.

Throughout my career, I have witnessed dozens of organization structure changes, all attempting to improve 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 on the organization chart, people trained, software purchased (and, in some cases, used), books of procedures—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. In retrospect, I realized that the one experience I had before learning of TOC that did result in significant change to the project delivery system addressed 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 did, the change would have been even more successful.

Now, having seen CCPM deliver in multiple organizations, I know why that earlier solution worked.

4.1.1 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 clarifies the system goal, identifies the system inputs and outputs, and leads to the measures that aid controlling the system to achieve the goal (see Figure 4.1).

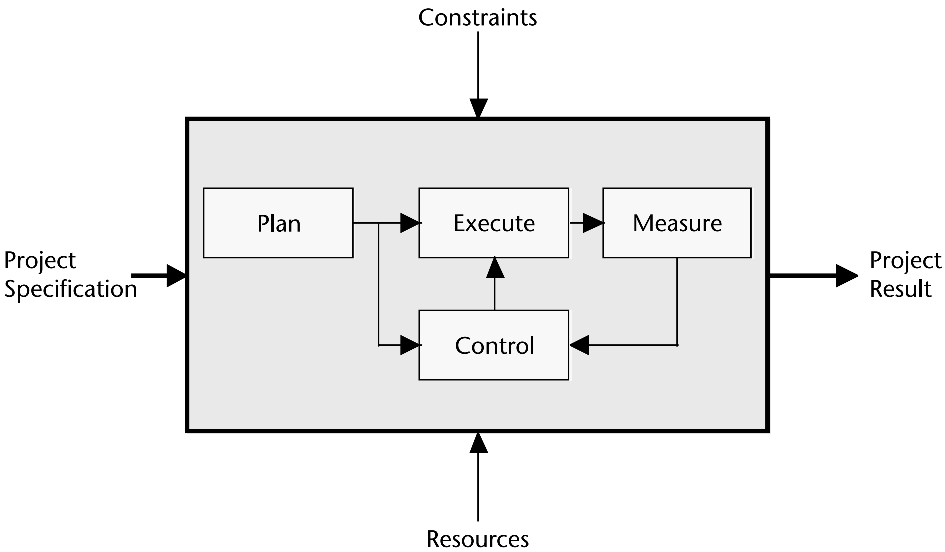


Figure 4.1 The black-box view of the project system process inputs to produce outputs that satisfy the system goal.

You can convert this black-box view to the Six Sigma SIPOC (supplier, input, process, output, customer) view by considering the stuff in the box as the project delivery system, the suppliers as those who provide resources to the system, and the customers as the project customers.

4.1.2 Project 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 a new hypothesis or approach. Undesired effects differ from the desired effects necessary to support the goal of successful projects. Undesired effects identified in Chapter 2 include:

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 become increasingly longer.
7. Many projects are cancelled before they are completed.
8. Project work creates high stress on many participants.

Undesired effects (UDEs) are things that we do not like about the present system. A good way to check them 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 that we are currently using. Even though they are not intended effects, persistence of the undesired effects 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 undesired effects. As the undesired effects are observed on 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 an underlying conflict or dilemma common to all of the environments causes these effects. To decide what to change, you have to first identify this dilemma: the constraint of the present system.

4.2 Identify the Constraint

Any project worth doing is worth doing fast. The main reason is that, for most projects, investment starts when the project starts, but return on investment does not start until the project is complete. A secondary reason is that the quality of the result can be much better if a project is completed as soon as possible by focused resources. Thus, the goal of a started project should be to finish as soon as possible, once started. We will consider the constraint to this goal.

Most projects performed today use the Critical Path Method (CPM). It was developed in the early 1950s and is taught as the centerpiece of most project management classes (I taught some classes for the University of Phoenix and University of Idaho). It is also described in all project management texts. Figure 4.2 illustrates a typical critical path schedule. The longest path through the network is the critical path.

Figure 4.2 also shows the resources assigned to perform each task as a number: resource number 1 and resource number 2. Assume that 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 the overall project. This project

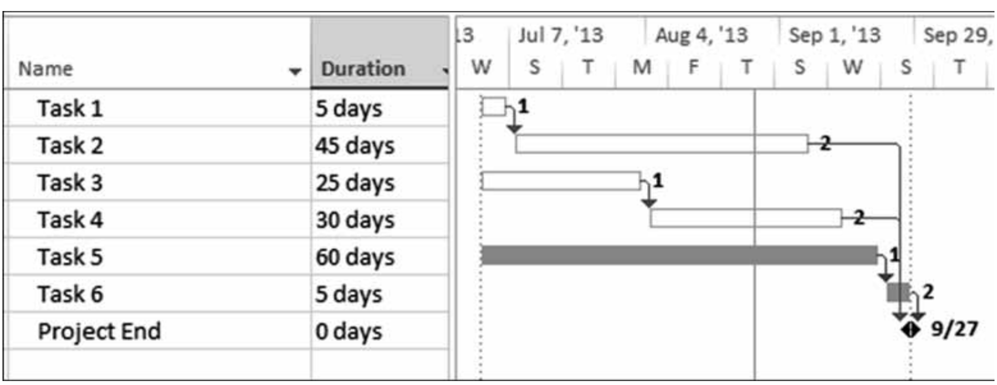


Figure 4.2 Example critical path schedule. (The numbers to the right of the bars represent resources.)

is planned to be late because only one task on the critical path taking longer than planned will cause the project to be late. Several will take longer than planned due to the planned multitasking. Nearly all CPM planned projects will be late if they did not account for finite resource availability in the plan because they hide conflicting resource demands, or, put another way, they assume infinite resource availability.

An associate asserted "our schedules assume infinite resources" to a group of engineering managers. They all disagreed strongly that they assumed infinite resources. Many of their projects did not have critical path schedules. Most of their projects that did have schedules did not identify the resources needed for the tasks. They nowhere summed the resource demand for all of their project schedules. They had no estimates for the demands from nonproject work, which in some cases comprised a substantial fraction of resource time. They had people assigned to multiple project tasks at the same time. They were confused by the fact that they did annual budget planning that accounted for resources. Because they allocated budget to resources, they thought that was resource planning.

Analyzing the project illustrated in Figure 4.2 allows us to estimate how long it might actually take. Consider the work performed by resource 1 (tasks 1, 3, and 5). As all three tasks are planned to start at the same time, this means each task will take three times as long when the three are being worked simultaneously. This condition will last for at least 15 days, 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, in effect, 5 planned 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 1 day per 2 days. Therefore, the remaining 20 days of work on task 3 will take 40 days, for a total task 3 actual duration of 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 for 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 4.3 illustrates the expected actual performance on the project, extending the date by over a month. The project was planned to fail.

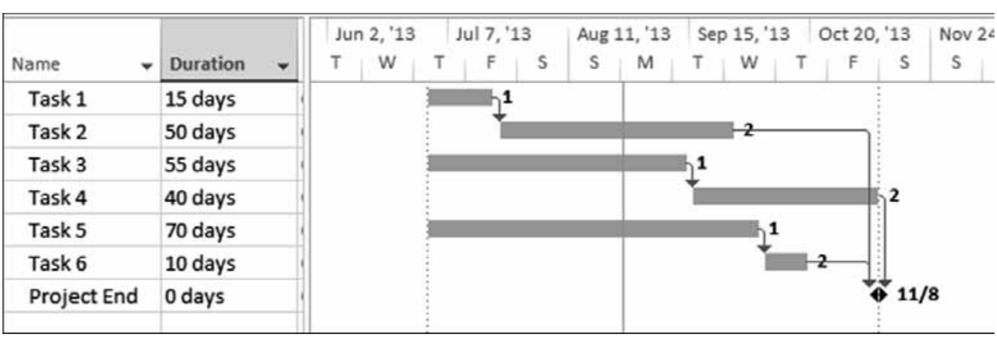


Figure 4.3 Actual task performance.

A method exists to solve this problem: resource leveling. Most CPM software includes the ability to resource level. Figure 4.4 illustrates the resource-leveled version of Figure 4.1. Note that the date of the resource leveled schedule conforms to the date of the actual schedule (Figure 4.3) resulting from accounting for multitasking. The resource-leveled schedule seems to resolve the first undesired effect: projects frequently overrun schedule. Of course, they do; they are planned that way.

A problem remains with the resource leveled schedule that I will address in some depth when discussing exploiting the schedule: accounting for variation in task performance. At this point, the schedules assume that task durations are deterministic (i.e., that you can predict an exact duration for each task). You know you cannot. That is why I stated that resource leveling only seems to resolve the first undesired effect. More is needed, but it is a first step in the right direction.

Examination of Figure 4.4 leads to an interesting observation. Note that the software identified the critical path as only comprising tasks 5 and 6. This is curious, because the software does not identify a critical path for the first part of the project depicted in Figure 4.4. It identifies the critical path during a backwards pass following linked tasks with zero float. It identifies the longest path so identified as the critical path. When the paths have gaps or tasks with positive float the calculation can proceed no further. The identified critical path after resource leveling will likely be different with different software because the resource-leveling algorithms differ. A simple 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 4.2 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 4.4), all of the paths have float or slack (space before the start of a chain counts as slack for this assertion). Thus, none of the paths is the critical path.

I do an informal survey in the classes I teach for the Project Management Institute. I ask how many resource load (i.e., identify the resources needed, as on Figure 4.2) their project schedules. Usually one-half to two-thirds do so. I then ask how many resource-level their schedules. 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 of the elite in project management; most are certified Project Management Professionals (PMPs).

I ask those who resource-load, but do not resource-level, "Why not?" Most do not have an answer, but those who do say one of two things:

1. The date moves out.

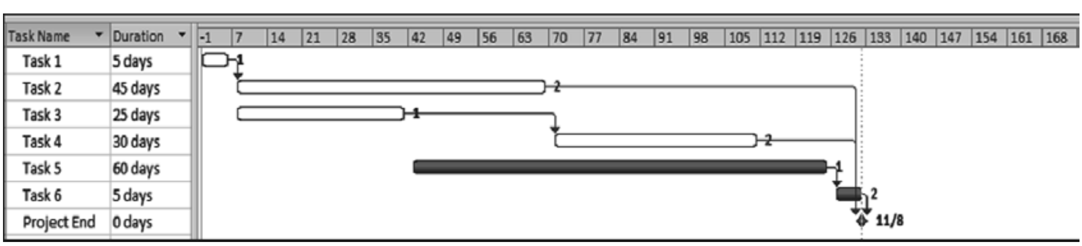


Figure 4.4 Resource-leveled critical path schedule.

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 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 4.5 illustrates the critical chain for our simple example project. It has no float or slack when defined.[1] Note also that it jumps the project logic paths (although it retains the technical logic paths in the project schedule).

In the past, I never questioned the proposition that an acceptable way to remove resource contention is to first 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 schedule 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 and 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.

Current project management software operates by starting with the activity structure (critical path), identified from the task data input by the scheduler, and only then considers the limited resources available for the project. Project management software identifies the critical path by using the links among the project activities, as input by the scheduler, and then calculates the longest time through the network of activities assuming no resource constraints. The Project Manager inputs resource availability. The software can then be directed to allocate 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 (allocation of resources to the activities with minimum slack 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

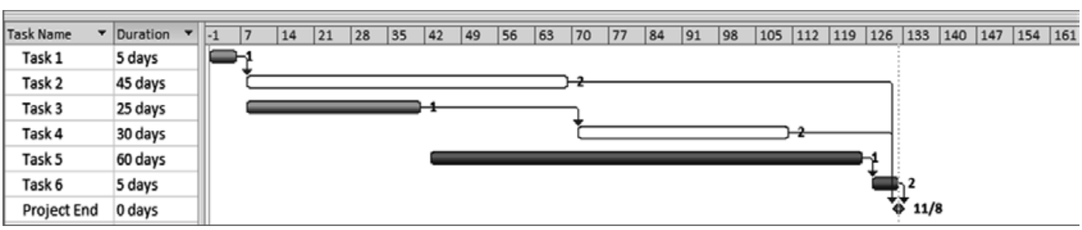


Figure 4.5 The critical chain is the longest path through the project after resource leveling.

1. The small gap between tasks 5 and 6 is a nonworking weekend that does not count.

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 constraints.

4.3 Exploit the Constraint

The original TOC thinking process method went directly from the undesired effects to create a current reality tree, a system model of the present reality that was causing the undesired effects. The procedure started with any two undesired effects (UDEs), although four UDEs were recommended for most cases, and built a logical connection among them with non-UDE entities. It then added one UDE at a time filling in the logic with non-UDE entities as needed until all of the undesired effects were connected in a system representation of current reality. After a process of what Popper would call critical discussion, the analyst selects an entity at the bottom of the tree as the core problem and proceeds to analyze it as the result of a conflict. This led to an initial change to begin the design of a new system that no longer creates the UDEs and in fact creates their opposing desired effects. This process worked. However, it was hard and lengthy.

A later innovation, which Dr. Goldratt indicated was suggested to him by someone else, made the process more direct and seemingly easier to operate by more people. This method selects three of the undesired effects 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 a future reality free of the undesired effects. I have reservations about using the process when analyzing a new problem but I find it a useful way to describe a completed analysis. Therefore, I will use it to address how to exploit the critical chain of a project.

4.3.1 Projects Durations Become Longer

Most people agree that projects seem to take longer than they were scheduled to take or than previous similar projects took. 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 with singling 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 the Figure 4.6 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

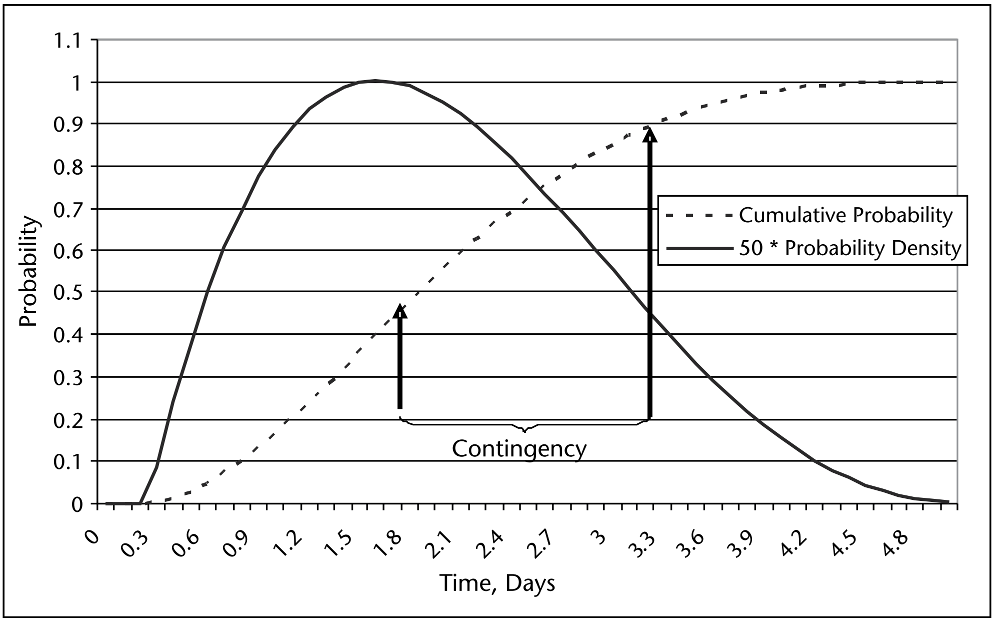


Figure 4.6 Variation in estimates for the time to perform a task helps define contingency.

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 using the word in communication have the same meaning.

Everyone wants to have a successful project. One necessary condition to have a successful project is to have the project complete on schedule. To have projects completed on schedule, we must have every task on the critical path complete on schedule. 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 critical path method. Further, as 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 consistently complete tasks on time are good performers and people who do not consistently complete on time are 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 that can complete sooner usually has an edge. Everyone knows that planning to complete the project sooner tends to reduce project cost, therefore helping make a competitive bid. For those performing research and development projects, the impact of a shorter development may make the difference between success and failure of the project. For deadline-driven projects, a shorter scheduled time usually alleviates the pressure to start now.

For all of these reasons, to plan a successful project, the Project Manager must have a shorter critical path for the project. To have a shorter critical path for the project, the Project Manager must have shorter task estimates that do not include contingency.

Figure 4.7 is the evaporating cloud for the dilemma I described above. Reality does not allow a single number to represent both 50% probable task estimates and high-probability task estimates so there is a conflict. In many environments, this conflict plays out by 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 have 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 have the same pressure to 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 towards a cliff or towards each other to see who would first "chicken out," 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 gets them, causing management to chicken out and extend the project time. This will give them the time they need to complete their task on time, 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 as a nonperformer or nonsupporter. They have no choice in the real world of power politics.

4.3.2 Projects Frequently Overrun Schedule

When asked why projects overrun schedule, people usually say that the projects start out fine but somewhere along the way a snag develops that begins to push one or more deliverables later and later. Everyone knows that it only takes one task to be late 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 diverting resources and making changes in the project schedule that cost more (e.g., overtime and extra resources). They focus on the part of the schedule that is slipping. The people working on the snag usually feel a lot of pressure to get their part of the

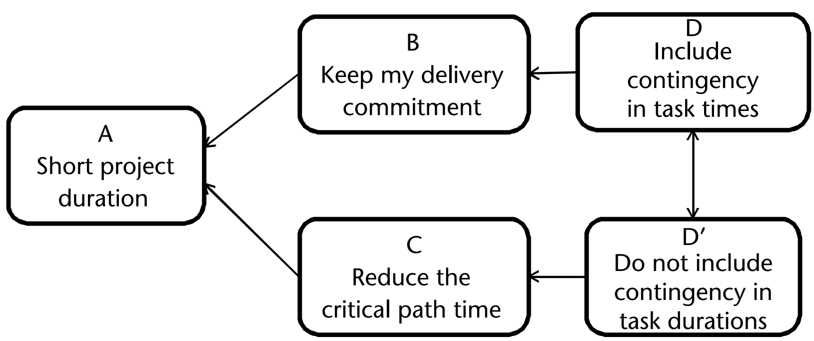


Figure 4.7 Task time conflict.

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 on which they were supposed to be working, 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 it on poor performance by the group responsible for that part of the project. The second type of response is more general, blaming it on the tendency of stereotype task performers to underestimate, or on management's setting arbitrary completion dates.

Parkinson's Law

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 that this occurs less frequently than you expect, if the estimates were truly 90% probable estimates. Even with skewed distributions, tasks should complete early a substantial percentage of the time. Figure 4.8 illustrates actual results for the ratio of actual task duration to estimated duration for over 3,000 tasks on an actual large project. The bars show the count of tasks and the line shows the cumulative frequency. It shows that most frequently tasks complete exactly on the due date: a ratio of 1. It also shows about 80% of tasks complete in that time or less, suggesting low-risk estimates. The second-highest frequency on this graph is tasks that took essentially no time. This represents tasks that did not really have to be performed. Other analyses that I have performed like this in two other organizations show as high as 80% of tasks complete on their due date. This is not consistent with the task completion time estimate distribution presented in Figure 4.6, suggesting Parkinson's law at work.

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

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 was believed, leaving little potential for positive variations.

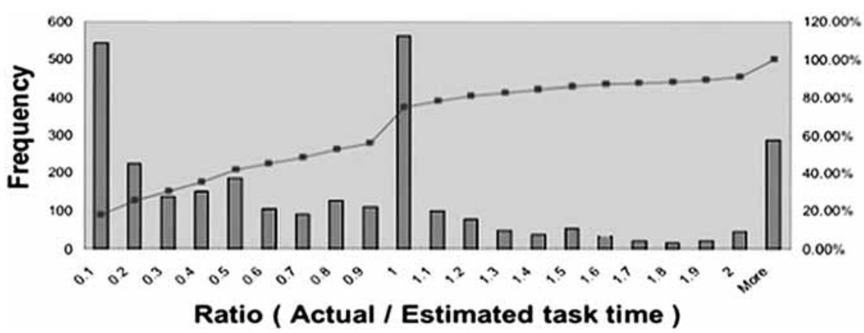


Figure 4.8 The distribution of actual task completion time differs from the estimate distribution. It shows a remarkable percentage of completions right on the due date.

3. The work expands to fill all available time and budget.
4. The belief exists that the next activity would not be ready to use the task result anyway.
5. In most organizations, there are significant penalties (threatened or real) for completing activities late. What are the "rewards" for completing activities early? Do you ever see any of the following:
   - Reduced budget for the performing organization leads to higher overhead rates and, in extreme cases, downsizing.
   - Reduced credibility in performers' 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 of the 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 any advantage of early completion or reward the task performers if they do deliver early. Paying the resource performing the work in accordance with the time they spend on the task motivates them to use up all of the duration authorized. If your project uses a cost reimbursement contract with task performers (the usual practice for resources in the company and for certain types of external resources), they may even be incentivized to slow down the work to get overtime pay or more total revenue from our project.

If one resource gets its activity done early, what is the chance that the next critical resource down the line is ready to start working on its activity? If it is a critical resource, it is in demand and has limits on availability. It does not seem likely that they will be able to work on our activity until the date they had planned for it. Therefore, we lose the positive variance and introduce wait time. This means that the actual schedule time grows due to activity dependence.

Critical Chain describes several effects that lead to performance systematically over-running task due dates although they initially had extensive contingency time (Goldratt, 1997). Meredith and Mantel state, "...operation of *Parkinson's Law* …clear and present danger. The work done on project elements is almost certain to 'expand to fill the additional time.'" In Dr. Goldratt's words, the safety time is wasted.

In the business and government cultures that I have witnessed, there is little or no reward for completing individual tasks early and 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 complete 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, they cannot continue to charge to the project. They must find alternative work for the resources. If individuals complete tasks early, they get more to do. These cultures drive local optima: 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 when it appears as extra time at the end of tasks. Eliminating task due dates removes the antecedent for this behavior.

This phenomenon is well known in other fields. It is called Parkinson's law after a book by that title (Parkinson, 1957). The statement of the law is, "The work expands so as to fill the available time for its completion" (p. 2). Setting dates for the completion of individual tasks is the antecedent for Parkinson's law behavior. The solution needs to eliminate both that antecedent and the reinforcement for the behavior.

All of this leads to the second conflict illustrated by Figure 4.9. The upper path refers to the performing resource. To be a successful team member, I must contribute to early completion of the project. To contribute to early completion of the project, I must turn work in early. On the lower branch, to be a successful team member, I must have sufficient time in my task estimates to complete my commitments. To complete my commitments, I must not turn in my task result 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 did finish early and turn it in to my manager to be checked prior to submitting it to the project, she would not likely look at it until it is due anyway because she is a very busy person.

Student Syndrome

Did you always study for your exams weeks ahead so 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 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 on whom the Project Manager is counting to get the critical path work done on time.

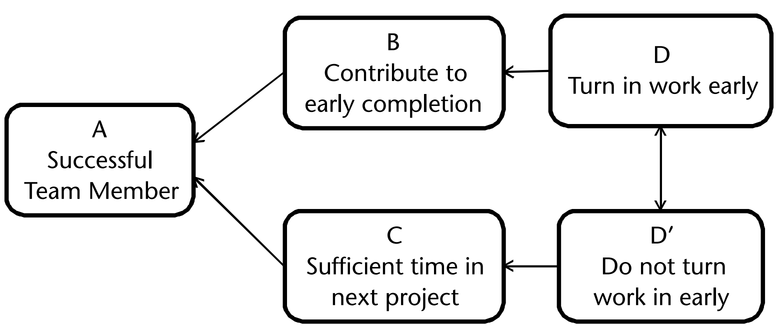


Figure 4.9 One conflict underlying project schedule overruns.

Figure 4.10 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 have a problem to complete the activity in the remaining time, during the first two-thirds 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 to keep to the activity duration by a little extra effort. What is the chance that they can recover from an unanticipated problem, like the computer crashes? This behavior is known as *student syndrome*.

Student syndrome behavior results in little chance of seeing the positive side of activity duration variation. The effects described above make it unlikely 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 most likely 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 noncritical 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?

Anchoring

A more technical description of the above two behaviors is to consider them both as instances of the psychological cognitive bias called anchoring (Kahneman, 2011). Much psychology research confirms the behavior. The mere fact of presenting a number to people causes behavior in the direction of that number. For CCPM the numbers are estimates of task duration or the finish dates of tasks in the schedule. Presenting a task finish date can cause either of the behaviors to try to complete the task on that date and not before. I call this date-driven behavior and it includes both the effects described anecdotally by Parkinson's law and student syndrome.

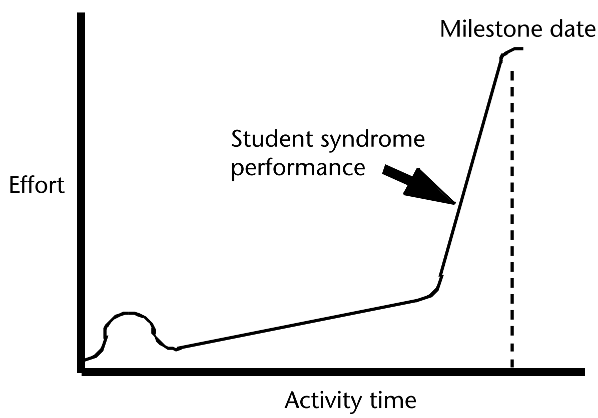


Figure 4.10 People perform most activities, and most people follow the student syndrome performance curve.

4.3.3 Multitasking

Now let us assume that the schedule system demands that resources start on activities as soon as possible and report the task start to the Project Manager. Let us assume the person splits their time during the day evenly to three assigned activities. 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, then none of the activities complete until the third week. Multitasking has increased the activity duration for all three projects to three weeks. They have delayed throughput on the first project for two weeks and on the second for one week.

The multitasking problem is actually much worse than just multiplying by the number of tasks for two powerful reasons. First, a growing body of research shows that each time one switches from working on one task to another task, some time is lost in making the switch and getting up to the previous level of productivity on the new task. Some research suggests this time can be 15 minutes or more for intellectual tasks. It might be even longer for physical tasks. The lost time can reach 40% of productive time. This lost time is pure waste. It is system capacity put to waste. Figure 4.11 illustrates a more realistic comparison of focusing versus multitasking.

The second problem with task switching is that errors often happen when people shift from one task to another. The organizations that have measured the quality defect rate before and after Critical Chain have shown quality defects to plummet, in some cases by 50%. Quality defects also require resource attention for repair, thus further reducing productivity.

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.

Dr. Goldratt demonstrated in *The Goal* how focus on local efficiency could damage the overall performance of a system. He used the example of robots, which

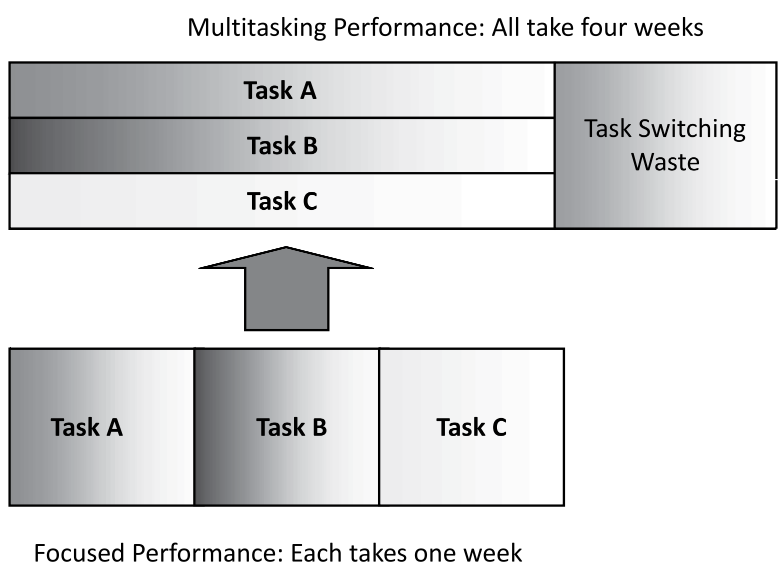


Figure 4.11 Multitasking delays all projects. It also justifies using the longer task times in future plans.

were operated all of the time to show high efficiency. In the case of production, this leads to producing excess inventory, and may "plug" the constraint with work not necessary for current orders, increasing operating expense and delivery times with no positive benefit to the company as a whole.

While most people acknowledge the facts above, many argue, "It's just not realistic to do otherwise. We have to satisfy multiple needs." They agree with logic demonstrating that multitasking is a very poor (perhaps the worst) way to meet multiple needs. They acknowledge that it deliberately lowers their personal throughput contribution. They will also agree 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.

If multitasking is a normal way of business in a company, three or four weeks becomes the normal task duration for this task. Performance data supports this inflated task duration. If this is a critical chain task, the practice directly extends the duration of the project. Most companies admit to encouraging extensive multitasking.

Peter Marris (1996) contended that behavior such as multitasking is a social effect of the more powerful using the less powerful to shield them from uncertainty. In other words, management takes advantage of the lower level resources in the organization by creating the pressure that leads to multitasking.

4.3.4 The Core Conflict Leads to Undesired Effects

You can combine the three conflicts to obtain the underlying core conflict leading to all three conflicts examined. Because the conflicts derived from the three starting undesired effects, resolving the core conflict should have a desirable impact on all three of the undesired effects analyzed. As the project system is a connected system, the core conflict may contribute to the other undesired effects as well.

Figure 4.12 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 or her own success. To have a successful project, each task must be performed as planned. 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 towards project success. To have the project succeed, each part of the project must contribute to overall project success. To contribute to overall project success, each task must subordinate to the overall project.

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

Figure 4.13 illustrates how the core conflict leads to all of the undesired effects. This implies that the core conflict is a high leverage part of the project system. A

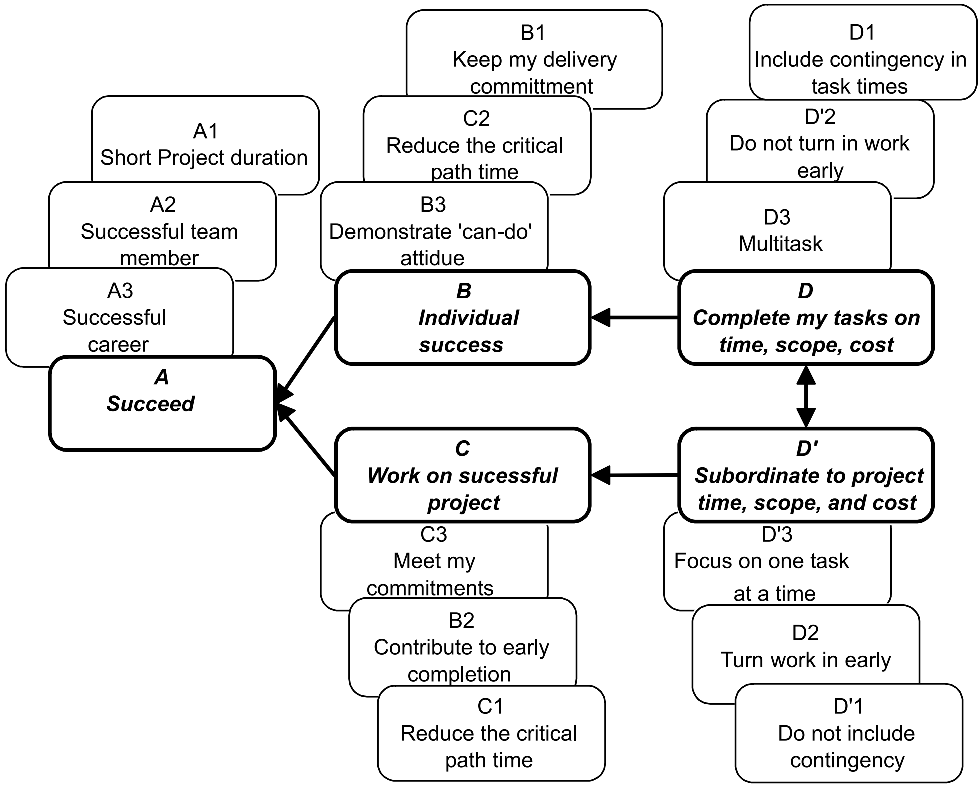


Figure 4.12 The core conflict underlies all three conflicts.

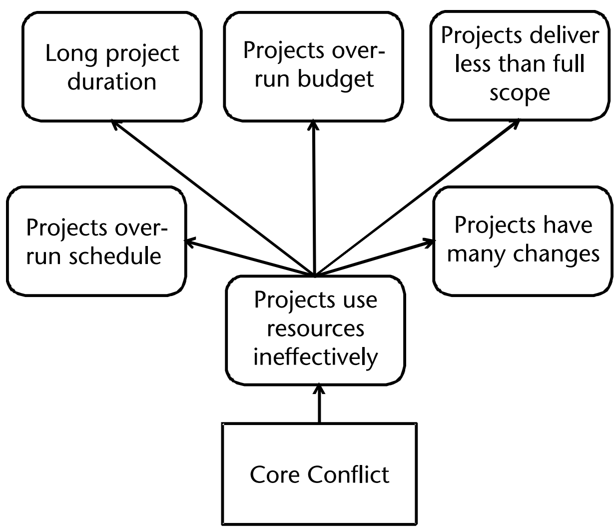


Figure 4.13 The core conflict leads to all of the system undesired effects.

solution (new theory) that resolves the core conflict differently can influence the whole system in a way that tends to move the undesired effects to their desirable counterpart.

The logic illustrated by Figure 4.13 is incomplete. It is only a notional connection between some part of the core conflict and the undesired effect. If you accept that the core conflict underlies most or all of the undesired effects of the project system, you may be willing to consider the beginning of the solution direction.

4.4 Towards Desired Effects

4.4.1 Resolving the Core Conflict

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

One element of resolving the core conflict must be to correct for the behavior elements: multitasking, Parkinson's law, and student syndrome. The solution to it is the first new rule: focus. The new project delivery system must enable people to focus on one task at a time until they complete it. They must be allowed to limit their personal WIP. While it helps if it is the right task to focus on (more on what that means later), when you eliminate multitasking, all of the tasks you were working on finish sooner than they would with multitasking because of eliminating the task nonworking time and the wasted time of task switching. So even if you work on tasks in the wrong sequence, all (on average) still complete sooner than they would have with multitasking. You will also make fewer errors.

Dr. 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* (Goldratt, 1984). He knew that, in most cases, the uncertainty in project duration estimates is much larger than the variation in production processes. He 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.

Goldratt describes the impact of variation and dependent events by the saga of Herbie in *The Goal*. He used the scenario of a troop of Boy Scouts on a hike through the woods. The trail is narrow, so the scouts cannot pass the one in front. 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 Boy Scouts is not the same. There are statistical fluctuations in how fast they walk. They are dependent on the scout in front of them, though, because they cannot pass. These fluctuations cause the length of the line to grow continuously. Herbie turns out to be the slowest scout, the constraint. The gaps in the line compare to inventory in a manufacturing plant.

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 behind a slower boy; he can catch up, but not pass. The line grows in length. This is worse than the manufacturing case. In manufacturing, they eventually use the inventory. In a project, we lose the time forever. There is no conservation of time.

The direction of the solution that Goldratt proposed 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 chain.

How then is the critical chain exploited? Dr. Goldratt was a Ph.D. physicist. He knew statistics and knew a lot about the "cloudy" behavior of much of reality. He knew that the only way to take advantage of our statistical knowledge is through dealing with numbers of events. Deming and Shewhart (Shewhart, 1986) before him had pointed out that science could not 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 his production solution where buffers of in-process inventory are strategically placed in front of machines to prevent them from running out of work. This solution implements Rule 2: Buffer.

Concentrating contingency in the buffer creates two significant bonuses. The first bonus is a shorter schedule. It is a mathematical fact the variances of the sum of samples from a series of independent distributions add. 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 variations. While attempting to protect the completion date of each task in a project, each task had to include its own allowance for uncertainty. These allowances add up along the path. When we take these allowances out of each task and put them at the end of the path they add as the square root of the sum of the squares, a much smaller total amount. Figure 4.14 illustrates how this works for a very simple case. The reason for this is evident. Some of the tasks should overrun, and some of them should underrun. The distribution of the sum need not be as large as the sum of the individual variations because some can cancel out if you can eliminate Parkinson's law and the student syndrome.

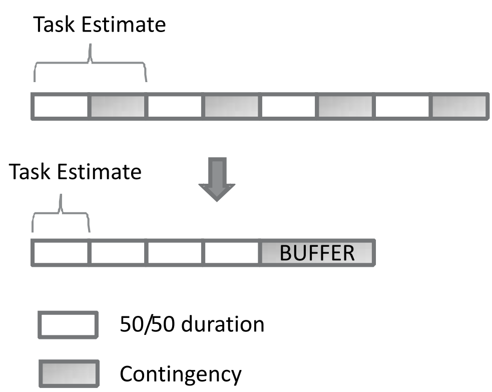


Figure 4.14 Concentrating contingency at the end of the path requires less total project time.

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 towards 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 uses average task completion times in the schedule and adds an aggregated buffer at the end of the schedule for overall project contingency. Mathematically, the mean for each task duration is the average duration that sums along a path. The project buffer is one part of the second of the three new rules introduced in Section 2.8.

4.5 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 CPM shows more content in the Critical Chain theory because it:

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

Popper (1976) noted that a new theory should contain and explain the old theory. 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, Critical Chain contains the critical path solution.

Popper suggested that the primary method of testing a new theory 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 stated (Popper, 1976):

> (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 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 it. However, none of the current solutions makes uncertainty management part of the basic project system in the manner of Critical Chain. 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. This is one way CCPM brings the power of Six Sigma and TQM to Critical Chain. Figure 4.15 illustrates the intention that the new solution direction will eliminate all of the UDEs with the current system.

Critical Chain applies buffers to resolve the reasons for schedule overrun caused by the reality of statistical fluctuations (uncertainty or variation) and dependent events. The CPM theory does not address this reality; it uses deterministic durations, starts, and stop dates for activity in the schedule. Combining this technical assumption with human behavior leads to schedule overrun. Schedule overrun leads to cost overrun and reducing 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. The resource is as necessary to complete the task as the completion of the predecessor task. The CPM assumes that an acceptable solution to the resource constraint is to first find the unrestrained critical path and then 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. We could find no references describing the reasoning behind this assumption. Goldratt found it easy to notice this implicit assumption because finite resource scheduling was an explicit assumption in the production system models with which he had worked.

Some users of CPM have asserted that the resource constraint need not be as real as the technical constraint. That is, for many pieces of work, you simply cannot change the task sequence; you have to put in the foundation before you

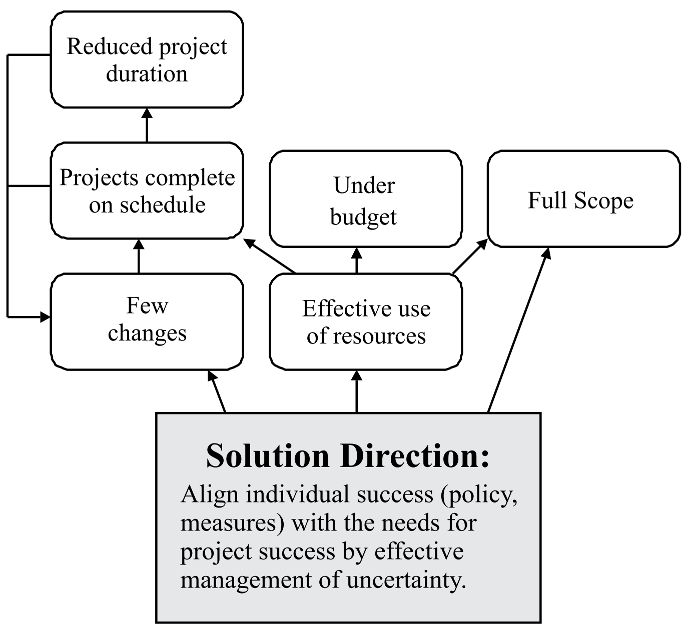


Figure 4.15 The new solution direction reverses all UDEs.

place the equipment, or you have to remove the equipment before you can reinstall it. However, one often can put resources on overtime, including extended work weeks, or elevate the resource through a variety of approaches. This does not conflict with CCPM, because the schedule 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 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 2 presented selected successful evidence that the Critical Chain method creates the desired effects. (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 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 11 addresses how to avoid this.

4.6 Multiproject System

The discussion above considered the direction of solution for a single project: the critical chain. However, there is not a critical chain for multiple projects. We have to reidentify the constraint to obtain the direction of solution for a multiproject system. Where the critical chain forms the constraint for a single project, something else must constrain the throughput of a multiproject system because there is no critical chain for multiple projects. Changing the constraint changes the direction of solution.

Considering the constraint of the multiproject system from the TOC perspective suggests that there must be a bottleneck to the flow of all of the projects. Most Critical Chain practitioners have come to call this the capacity constraint. It was originally thought of as a single resource shared across the multiple projects. If all of your projects share a common limited-supply resource, that can still be an acceptable way to think about it. The large variety of project systems that do not share a common resource across all the projects has led to a different way of thinking about the multiproject constraint. Now there is some agreement that the constraint is simply too much WIP: too many projects have been started and not finished. Excess project WIP also causes pressure on resources to multitask only now on tasks from multiple projects. Also keep in mind that nonproject work puts demand on resources, further increasing the pressure to multitask.

Little's law (Little, 1961) identifies the relationship between the throughput (T) of a system, the WIP (W), and the cycle time (C) for individual projects through the system. Although well known for production system, the applicability of Little's law to multiproject systems came from practitioners of Critical Chain, likely because these practitioner followers of Eli Goldratt had their background in production systems. Little's law is:

$$T = W/C$$

While Little's law is quite simple (and obviously dimensionally correct), using it is not intuitive. The equation suggests that the way to increase T is to increase W. Just the opposite is true. Lean manufacturing found that increased production and flexibility occurs with reductions in W: just-in-time production flow with an ultimate goal of single-piece flow.

Little's law is a single equation containing three unknowns. You may recall from algebra that solving for three unknowns requires three equations. In this case, the equation of most importance is the one that relates W and C. In contrast to the linear nature of Little's law, the relationship between W and C is very nonlinear. That is, over some ranges, small changes in W can cause large changes in C. Queuing theory provides just such a relationship. Figure 4.16 illustrates the queuing curve with the axes relabeled to illustrate the relationship between W and C. As Little's law derives from queuing theory, this seems like an appropriate relationship.

Examination of the figure illustrates that as WIP increases, the cycle time increases nonlinearly, up to infinite cycle time at a WIP equivalent to a resource utilization of 1. At relatively high levels of utilization (the right side of the curve), small changes in utilization yield large changes (reductions) in cycle time. Thus, reduction in WIP appears to be the appropriate direction of solution for the multiproject system.

The solution to WIP reduction is to Pipeline: the third new rule (see Section 2.8). Lean practitioners call it demand leveling. While the critical chain performed

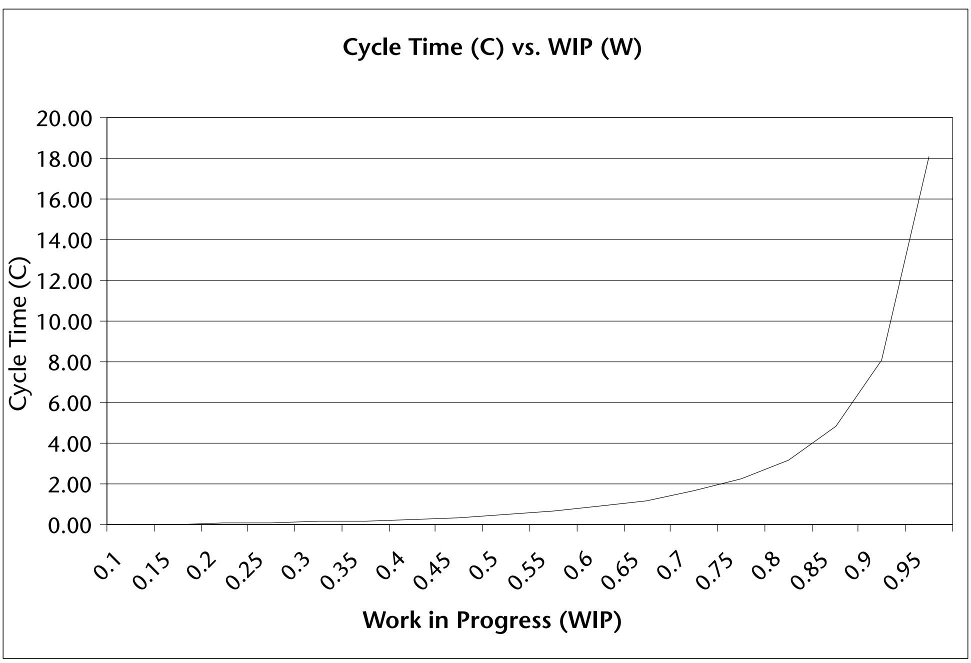


Figure 4.16 The relationship between WIP (WIP/Capactiy) and cycle time (*C*) in a project system.

demand leveling within a single project, pipelining causes demand leveling at the multiproject level. Chapter 8 continues the discussion for the multiproject system.

4.7 Execution

While the above discussion addressed human behavior issues during project execution and proposed a direction of solution to enable effective focus on one task at a time during execution, it does not directly enable that focus during execution because variation will cause actual demand to fluctuate compared to baseline schedules created with the above solutions. Thus, the full solution requires tools to enable focus on the right task during execution.

4.8 Determine What to Change to

Resolving the core conflict provides a necessary change to the system to begin to move towards the desired effects. The desired effects for the project system, derived from the current reality undesired effects, are:

1. Projects always complete on or before the scheduled completion date.
2. Projects complete within their budget (and sometimes show cost reduction).
3. Projects always deliver the full scope.
4. Projects have few changes.
5. Projects have 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.
9. Project workers are enabled to focus on one task at a time and complete it before moving on to another task.

The changes to resolve the core conflict provide a method to manage uncertainty and enable focused work. The changes acknowledge the reality of the resource constraint that affects many projects. The changes are not sufficient by themselves to create all of these desired effects. 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 single and multiple project solution leading to all of these desired effects and the full solution to achieve them.

4.9 Summary

This chapter identified the core conflict for project management systems as the way the project system manages uncertainty. Specific points made in developing that theory are:

- Undesired effects 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.
- The uncertainty management conflict manifests as the core conflict between protecting each task in the project schedule versus protecting the entire project.
- We have identified the constraint of a multiproject system as a resource shared across the multiple projects: the capacity constraint.
- The direction of the solution should be to enable focused work (Rule 1: Focus).
- The solution direction should manage uncertainty through concentrating contingency into buffers at the end of chains of tasks (Rule 2: Buffer).
- For the multiproject system, the appropriate direction of solution reduces WIP (Rule 3: Pipeline).
- A growing body of empirical evidence demonstrates the feasibility of the Critical Chain Project Management (CCPM) 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 manage, but they are not sufficient. You are managing only when you take actions that drive the system to the goal. Chapter 5 derives the full system to do that for single projects and Chapter 8 extends it for multiple projects.

References

Goldratt, Eliyahu M. (1984). *The Goal*. North River Press, Great Barrington, MA: North River Press.

Kahneman, D. (2011). *Thinking Fast and Slow*. New York: Farrar, Straus & Giroux.

Little, J. (1961). "A Proof for the Queuing Formula: $L = \lambda W$." *Operations Research*. 9 (3): 383–320.

Marris, Peter. (1996). *The Politics of Uncertainty*. London, U.K.: Routledge.

Parkinson, C. Northgate. (1957). *Parkinson's Law*. New York: Buccaneer Books.

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

Meredith, J., and Mantel, S. (1995). *Project Management: A Managerial Approach*. New York: John Wiley & Sons, Inc.

Shewhart, W. (1986). *Statistical Method From the Viewpoint of Quality Control*. New York: Dover Pulbications.

CHAPTER 5

The Complete Single-Project Solution

This chapter describes the process to develop the single-project management system to satisfy the system requirements identified in the previous chapters and further detailed in later chapters. Although presented as a forward-moving process from requirements to design, the actual process, as in nearly all designs, was iterative. Various design solutions were proposed and tested against the requirements until a suitable working system resulted.

5.1 From System Requirements to System Design

5.1.1 Requirements Matrix

Table 5.1 illustrates the requirements for an effective project management system following the method of Joseph Juran (Juran, 1988). Table 5.1 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 that I use in defining Critical Chain Project Management (CCPM). Many advocates of TOC and Critical Chain do not deal with the full scope of these requirements. Some are simply not aware of the full scope of requirements that the project system must meet and focus only on the part that Dr. Goldratt addressed. Others may not think they are necessary. I believe some organizations fail to achieve their full potential with CCPM because they do not satisfy all of the necessary requirements for effective project delivery. A system that meets the requirements of Table 5.1 will satisfy all necessary conditions for many projects although your project may have special requirements that go beyond this table. If so, I invite you to add them and continue on with the process to ensure your system meets your unique 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 would generate an identical list of project requirements. This list includes elements of the PMBOK™, elements of my own experience, and

Table 5.1 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*
1. Define the project system.	1.1 Project the system goal.	1.1 Define the project system goal to complete projects that make money for the company, now and in the future (for-profit companies).
	1.2 Define the project system boundary (what is a project?).	1.2 Define the project system boundary starting with customer needs and ending with a satisfied customer (see Section 6.7.1).
	1.3 Account for understanding of variation.	1.3.1 Account for common-cause variation in project processes.
		1.3.2 Provide a means to separate and deal with special-cause variation.
	1.4 Use the Theory of Constraints to design the system.	1.4.1 Identify the project constraint.
		1.4.2 Exploit the project system constraint.
		1.4.3 Subordinate everything else.
	1.5 Include knowledge of psychology in the system design.	1.5.1 Align project system needs with individual psychological needs.
		1.5.2 Align individual rewards with project system needs.
		1.5.3 Overcome task due date focus to reduce Parkinson's law and student syndrome behavior.
	1.6 Enable continuous improvement of the project system (a theory of knowledge).	1.6.1 Define and standardize processes.
		1.6.2 Measure process performance.
		1.6.3 Assess process performance.
		1.6.4 Improve processes.
2. Deliver the project result to the specification (scope).	2.1 Deliver all of the specified features.	2.1.1 Satisfy all of the physical requirements for the specified features.
		2.1.2 Satisfy all of the functional requirements for the specified features.
		2.1.3 Satisfy all of the operational requirements for the specified features.
	2.2 Satisfy all of the feature quality requirements.	2.2.1 Satisfy all of the feature quality requirements
3. Deliver the project result on time (schedule).	3.1 Deliver the project result on time (schedule)	3.1.1 Complete the project on or before the quoted completion date.
		3.1.2 Complete intermediate milestones on or before the quoted completion dates.
4. Deliver the project result for the estimated cost.	4.1 Total cost	4.1.1 Complete the total project within the approved budget.
		4.1.2 Do not spend more than specified maximums on subcategories of the total cost.
	4.2 Satisfy project cash flow requirements.	4.2.1 Do not exceed project estimated cash flow requirements.

elements specifically derived from the solution I am 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 Goldratt and considering the basis of Critical Chain, I would not have included accounting for common-cause variation among project requirements.

Table 5.1 (continued)

Primary Requirement	*Secondary Requirement*	*Tertiary Requirements*
5. Satisfy unique individual project stakeholder needs (in addition to the above).	5.1 Satisfy project client.	5.1.1 Solicit and specify all requirements necessary to deliver a satisfactory final product.
		5.1.2 Provide evidence of meeting the project specifications.
		5.1.3 Provide information during the project to enable decisions that may affect the balance of the project.
		5.1.4 Respond to requests for changes.
		5.1.5 Visually control the project.
	5.2 Satisfy project team.	5.2.1 Clear scope definition including assumptions.
		5.2.2 Designate responsibility and authority assignment.
		5.2.3 Develop a Project Plan specifying who has to do what by when.
		5.2.4 Feedback to control performance to plan.
		5.2.5 Define the method to control interfaces with other team members.
		5.2.6 Provide the method to raise and resolve issues during project performance.
		5.2.7 Operate a change control process.
		5.2.8 Provide visual control of tasks.
		5.2.9 Develop the method to enable resources to focus on one task at a time (control WIP).
	5.3 Satisfy subcontractors and supporting resources.	5.3.1 Clarify scope definition.
		5.3.2 Identify clear responsibility and authority assignment.
		5.3.4 Develop a Project Plan specifying who has to do what by when.
		5.3.5 Feedback to control performance to plan.
		5.3.6 Operate a change control process.
6. Handle intrusions of nonproject work.	6.1 Ensure completion of nonproject work.	6.1.1 Design a system to prioritize nonproject work with project work.
		6.1.2 Provide a process to promote nonproject work for execution.
	6.2 Prevent multitasking	6.2.1 Process to limit overall WIP.
		6.2.2 Process to prioritize nonproject work versus project work.

The table of project requirements can never be complete. It is a 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 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 2.

5.1.2 Summary of Single-Project Critical Chain

Figure 5.1 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 chain through the project considering both the task logic and the resource constraint.
2. Removing resource contention from the project schedule 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 5.1 illustrates the buffer as a shock absorber.)
4. Subordinating merging chains with feeding buffers (while continuing the elimination of resource conflicts).
5. Ensuring resource availability especially for tasks along the critical chain (not illustrated in Figure 5.1 but described later).
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 effectively use single project CCPM. The following list indicates which of the new rules (introduced in Section 2.8) the behavior supports. I cover Rule 3, Pipeline, later because it applies to the multiproject approach. The behavior changes are:

1. Management encourages using mean task duration estimates by not pressuring people to perform to the estimated durations or task dates and ensuring adequate buffering in schedules (Rule 2: Buffer).
2. Management enables people to focus on one task at a time (Rule 1: Focus).

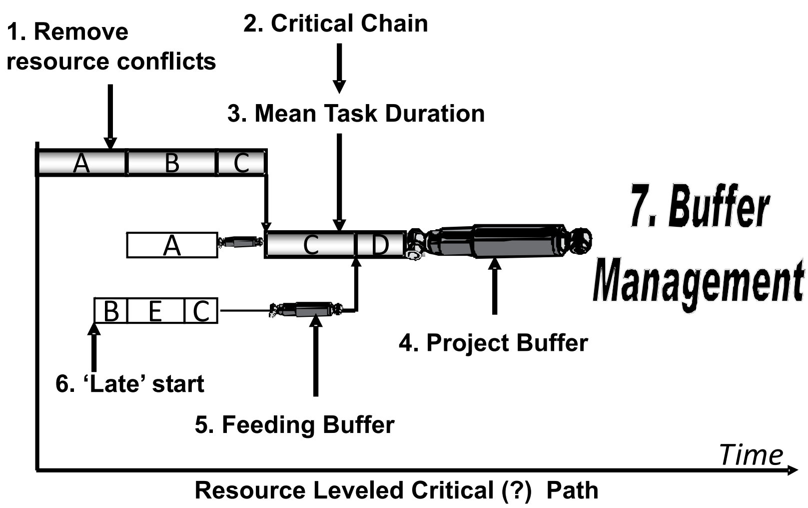


Figure 5.1 Key features of the Critical Chain solution deliver performance to the project system requirements.

3. Resources focus on one task at a time and pass on the results as soon as they complete the task (Rule 1: Focus).
4. Management statuses the schedule frequently with task actual start and finish dates and remaining duration estimates for tasks started but not finished (Rule 2: Buffer).
5. Everyone uses the schedule and the buffer reports to decide what to work on next (Rule 2: Buffer).
6. Management uses the buffer status to decide when and where to take action on the project to ensure success (Rule 2: Buffer).

The behavior changes are what cause the huge improvement in project system performance with CCPM. Creating critical chain schedules without management leadership of the behavior changes accomplishes little.

5.2 Developing the Critical Chain Solution

The following sections describe the single project Critical Chain features in terms of the Theory of Constraints (TOC) focusing steps. I do not know if this is how Dr. Goldratt derived these features. Following Popper's description (Popper, 1972) of a theory of knowledge and the scientific method, how Goldratt defined these features (which Popper would have called bold conjectures) does not matter. Instead, it matters that we subject the conjecture to critical discussion and test to see if the discussion supports the selection of Critical Chain over CPM.

5.2.1 Identifying the Project Constraint

As most projects do not create benefits until they are complete, the evident constraint of a project is the chain of tasks that takes the longest to complete the project. 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 Critical Path Method (CPM) as:

> A schedule network analysis technique used to determine the amount of scheduling flexibility (the amount of float) on various logical network chains 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.

These definitions contain several important but unstated assumptions. First, there is an assumption of a single (deterministic) task duration. Second, there is no information regarding the probability one should attach to that duration. Is it a 50–50 duration or is it a high-probability duration? The differences are very significant and crucial.

The PMBOK Guide includes discussion of probabilistic scheduling approaches, including PERT and Monte Carlo simulation. Explanations of these approaches are too brief to be directly usable. My surveys indicate they are little used in practice and, without further elaboration, do not help project execution. They are just project network analysis methods.

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 chains.

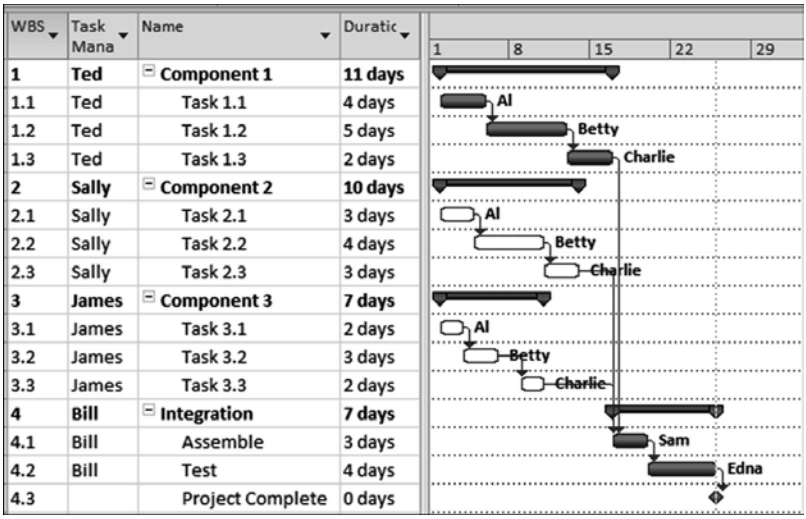
The basic definition of the *critical chain* is to simply identify the constraint of the project or, "The sequence of dependent events that prevents the project from completing in a shorter interval. Resource dependencies determine the critical chain as much as do task dependencies."

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 Method (CPM) project scheduling contains a hidden assumption that an acceptable way to account for potential resource constraints on the project is to first 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. For most networks, the application of the resource-leveling algorithms lengthens the overall schedule. This may be one reason that few project schedulers use the resource-leveling tools.

I have conducted an informal survey while delivering lectures to members of the Project Management Institute. These groups include many certified Project Management Professionals (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 not, those who respond most often state that resource leveling causes the plan length to exceed management demands.

Figure 5.2 illustrates a typical critical path project schedule showing the resources needed next to each task bar. Most Gantt charts that I have found in use do not show the resources and most of the schedules I have examined do not identify the necessary resources. There are notable exceptions. Hopefully you are one of the exceptions. For this project, we would fail to meet the schedule on the project because each resource can only do one task at a time and several resources are scheduled to work on two or three tasks in parallel. We can see this because the resources are shown and because it is a very simple schedule. We might not know this if the resources were not shown. We could not know it if the resources were



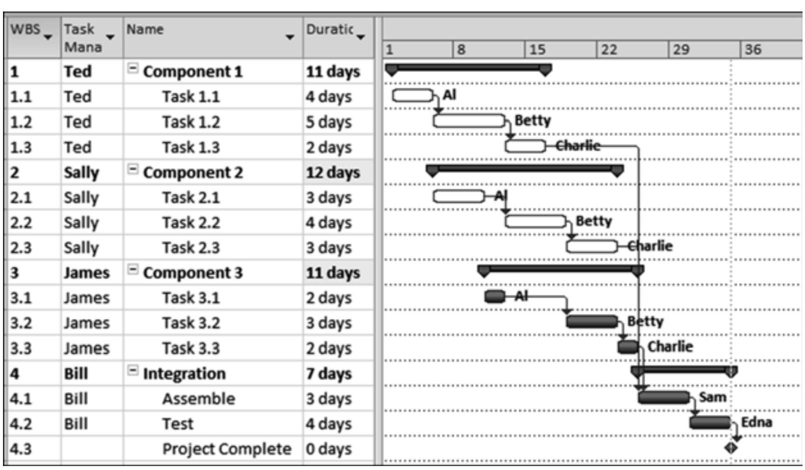
WBS	Task Mana	Name	Duratic
1	**Ted**	**Component 1**	**11 days**
1.1	Ted	Task 1.1	4 days
1.2	Ted	Task 1.2	5 days
1.3	Ted	Task 1.3	2 days
2	**Sally**	**Component 2**	**10 days**
2.1	Sally	Task 2.1	3 days
2.2	Sally	Task 2.2	4 days
2.3	Sally	Task 2.3	3 days
3	**James**	**Component 3**	**7 days**
3.1	James	Task 3.1	2 days
3.2	James	Task 3.2	3 days
3.3	James	Task 3.3	2 days
4	**Bill**	**Integration**	**7 days**
4.1	Bill	Assemble	3 days
4.2	Bill	Test	4 days
4.3		Project Complete	0 days

Figure 5.2 The critical path does not account for the resource constraint.

not estimated for the tasks. Not accounting for the potential resource constraint is a major reason many projects fail on schedule: they are planned to fail by not leveling the resource demand before starting.

Figure 5.3 moves tasks to eliminate the overlap of resource demands. In a manner similar to many computer algorithms for resource leveling, the schedule first gives the resource to the path with least float which is usually the initial critical path. Note that when leveling is done, all chains have float (or slack) so that there is no critical path defined as the chain 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 not checked what the software then does 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



WBS	Task Mana	Name	Duratic
1	**Ted**	**Component 1**	**11 days**
1.1	Ted	Task 1.1	4 days
1.2	Ted	Task 1.2	5 days
1.3	Ted	Task 1.3	2 days
2	**Sally**	**Component 2**	**12 days**
2.1	Sally	Task 2.1	3 days
2.2	Sally	Task 2.2	4 days
2.3	Sally	Task 2.3	3 days
3	**James**	**Component 3**	**11 days**
3.1	James	Task 3.1	2 days
3.2	James	Task 3.2	3 days
3.3	James	Task 3.3	2 days
4	**Bill**	**Integration**	**7 days**
4.1	Bill	Assemble	3 days
4.2	Bill	Test	4 days
4.3		Project Complete	0 days

Figure 5.3 Removing resource conflicts usually creates gaps in the critical path.

the Theory of Constraints method of project planning always considers it. Thus, the critical chain includes the resource dependencies that define the overall longest chain (constraint) of the project. The method resolves all resource constraints while determining the project critical chain. The project critical chain will not have gaps between tasks at this point. It is the longest chain through the project considering both the task logic and resources. Figure 5.4 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 chain as the critical path. This is an important fact in verifying the integrity of the Critical Chain method; it contains the critical path method 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 critical path stated that the critical path may change during the performance of the project. This can occur whenever project tasks experience common-cause variation that redefines the longest zero float chain to complete the project. Due to our knowledge of variation, this means that we should expect the apparent critical path to change frequently. Dr. 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 implementation in many project management systems institutionalizes this mistake. This does not enable the project team to focus on the constraint to the project but instead causes them to make the error of chasing an ever changing critical path. As Dr. Deming illustrated with his funnel experiment, this will always make the project system perform 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 schedule construction procedure and the subordination step described later. Instead of changing the critical chain once the project moves into execution, we will determine task priority by the effect tasks are having on the buffer. This method enables integrating task priority across multiple projects.

WBS	Task Manager	Name	Duration
1	**Ted**	**Component 1**	**11 days**
1.1	Ted	Task 1.1	4 days
1.2	Ted	Task 1.2	5 days
1.3	Ted	Task 1.3	2 days
2	**Sally**	**Component 2**	**12 days**
2.1	Sally	Task 2.1	3 days
2.2	Sally	Task 2.2	4 days
2.3	Sally	Task 2.3	3 days
3	**James**	**Component 3**	**11 days**
3.1	James	Task 3.1	2 days
3.2	James	Task 3.2	3 days
3.3	James	Task 3.3	2 days
4	**Bill**	**Integration**	**7 days**
4.1	Bill	Assemble	3 days
4.2	Bill	Test	4 days
4.3		Project Complete	0 days

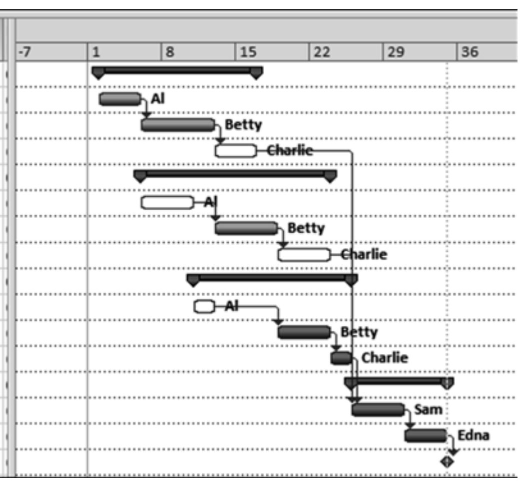


Figure 5.4 The critical chain includes both the resource and task logic constraint to completing the project on time or sooner.

5.2.2 Exploiting the Constraint

Having defined the critical chain as the initial 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 Dr. Goldratt's unique focus on statistical fluctuations and dependent events leads to a significant departure from most current project systems. Dr. Goldratt's recognition of variation is not unique but applying his buffer solution to project management was an innovation.

Dr. Deming noted that managers often make systems worse by not understanding the fundamental difference between common-cause and special-cause variation. He also noted, "I should estimate that in my experience most troubles and most possibilities for improvement add up to propositions something like this: 94% belong to the system (responsibility of management), 6% special."

Projects have common-cause variation in the performance time of tasks. Although the time 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 Theory of Constraints improvements for production take advantage of (exploit) the reality of statistical fluctuations and dependent events. Figure 4.6 illustrated a typical project task performance time distribution. The solid curve (left ordinate) showed the probability of a given time on the abscissa. The dotted line showed 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 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 common-cause variation using three task duration estimates but for a variety of reasons never caught on. The PMBOK Guide and literature still mention PERT in this fashion although it is little used today. "PERT diagrams" referred to in much of the project literature and in many project software packages are simply a way to show the project network logic independent of the time scale; 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 such as Parkinson's law, the student syndrome, and multitasking. Buffers monitor common-cause variation enabling CCPM Task Managers to use a prioritized task list to help resources focus on the right tasks when they are available. This enables Rule 1: Focus.

Exploiting Project Task Estimates

CCPM seeks to explicitly use mean (approximately 50% probable) individual task time estimates. The CCPM Project Manager recognizes that actual individual task performance times include common-cause variation and does not criticize task performers for individual task duration performance. Instead, the Project Manager encourages focus on one task at a time and turning it in as soon as it completes. This supports Rule 1 in execution.

As noted in the previous chapter, most Project Managers implicitly attempt to account for individual task common-cause variation by adding contingency time into each estimate, but 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% completion on or less than the task duration estimate. Figure 4.6 illustrated 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, stated (Shewhart, 1986):

> 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 for a project. I know that to be true but understanding variation changed my understanding of what it 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 clarified 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 it. Thus, most people misunderstand or at least misstate the meaning of accuracy.

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 of the to the center of the target. Of course, you are not really measuring the accuracy of the gun in that case either.

You are measuring the accuracy of the gun, shooter, cartridge, and environment. Changing any of them will change the apparent accuracy of the gun. For example, letting my son shoot instead of me will make a much more accurate gun. Shooting at 25 yards versus 100 yards will make 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 estimate. 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 cause a smaller range of 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 will tend to forget predecessors leading to the outcome (recall the student syndrome). This mental feat has two interesting effects:

1. The Project Managers selectively remember the instances where task duration estimates were exceeded, and therefore want to add contingency of their 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 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, my observations and study data indicate that a substantial portion (e.g., about one-third of IT 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 late project completion. (The stadiums in Atlanta and Greece caused anxious moments but were ready.)

Milestone performance in a very large project demonstrates that the task performance data conform very closely with Dr. 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. The very large project that I examined consisted of about 30 large subprojects, some of which contained yet smaller subprojects.

My experience shows project schedules from a variety of organizations (numbering in the hundreds of schedules) either fail to specify what probability and confidence of estimate is expected for task duration estimates and/or fail to provide a quantitative basis for the estimate. 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* (Kiley, 1996) 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 have ranges of several tens of percentages of the cost estimate. Therefore, in many cases, they have the same potential impact on schedule.

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 schedule but have no work assigned to them.

In the statistical terminology variance is the square of the standard deviation usually represented by s^2 or the Greek sigma squared. 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. Consider a chain of four tasks, each of which has a 50% probable estimated duration of one time unit, and a 90% probable estimated task duration of two time units. If you include the contingency in each task, the chain of tasks is eight units long. If you use the law of aggregation, you can protect the whole chain to 90% probability by scheduling the individual tasks at their 50% estimates (a total of four units), and adding a two-unit buffer at the end of the chain, for a total of six units.

A second factor that comes into play in aggregating tasks is the central limit theorem. 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.

Figure 4.14 illustrated 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 scheduled time of eight weeks. Based on the student syndrome and date-driven behavior, we consider it is unlikely to be delivered significantly before eight weeks and most likely to be delivered later.

The Critical Chain schedule 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 schedule including a two-week buffer is only six weeks. Considering date-driven behavior, there is some chance that the Critical Chain project will complete in four weeks. This compares to an eight-week schedule without a buffer. Adding buffers this way creates the nonintuitive result of shortening the project schedule. This is Rule 2 in scheduling: Buffer.

Exploiting Resource Availability

One of the leading alleged causes of late projects is that resources are not available or not available in sufficient quantity when needed. CCPM requires a mechanism to prevent critical tasks from starting late or taking longer than estimated due to resource availability. It does this by providing Task Managers or resources prioritized task lists for a short time into the future. The task priority is determined by the impact the task currently has on the buffers (more on this later). Figure 5.5 illustrates a simple example.

Goldratt proposed a resource buffer as an information tool to alert the Project Manager and performing resources of the impending necessity to work on a critical chain task. The prioritized task list for the Task Manager (as will be discussed later, this is the person accountable for task completion, usually the first-level supervisor of the resource performing the task) and/or resource performs this function. Note that inherent in the Critical Chain idea is that you cannot deterministically schedule resources. Because each task performance will vary, any forward deterministic schedule is an uncertain estimate. Each Resource Manager can establish the lead time necessary for their resources and use the project measurement and control process to alert the resource as the time of actual task performance approaches. Subcontractors are a resource. Enter them into the schedule program as you would any resource. You can also make the subcontractors contact the Task Managers and provide them with their tasks lists.

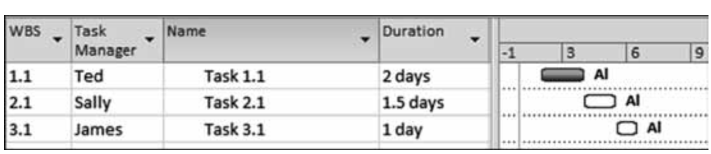


Figure 5.5 A prioritized task list provides notification of roughly when tasks will be ready to work and their relative priority.

You may choose to use alternative methods to motivate subcontracted resources to deliver when needed such as contract rewards or penalties for delivering to a specified lead time or duration.

5.2.3 Subordinating Merging Paths

Most projects have multiple chains of tasks. All task chains must merge into the critical chain by the end of the project if for no other reason than to merge into a milestone that identifies project completion. Usually, the chain 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 complete the last 10% in the second year." Figure 5.6 illustrates the filtering effect of merging paths. The successor task cannot start until the latest of the predecessor tasks is complete.

Task chain merging creates a filter that eliminates positive fluctuations and passes on the longest delay. The reason is that merging task chains means that all of the feeding chains 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 chain requiring three separate inputs to start. This occurs frequently 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 almost 88%. Even if each individual task had a 90% probability of completion, the probability of at least one being late is nearly 30%, or nearly one out of three times.

CCPM protects the critical chain from potential delays by subordinating critical chain feeding chains: placing an aggregated feeding buffer on each chain that feeds the critical chain. Figure 5.7 illustrates the placement of the feeding buffers. This includes chains 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 5.7 illustrates how the buffers absorb the late chains.

This innovation immunizes (to an extent) the critical chain from potential delays in the feeding chains. It also provides a means to measure the feeding chains

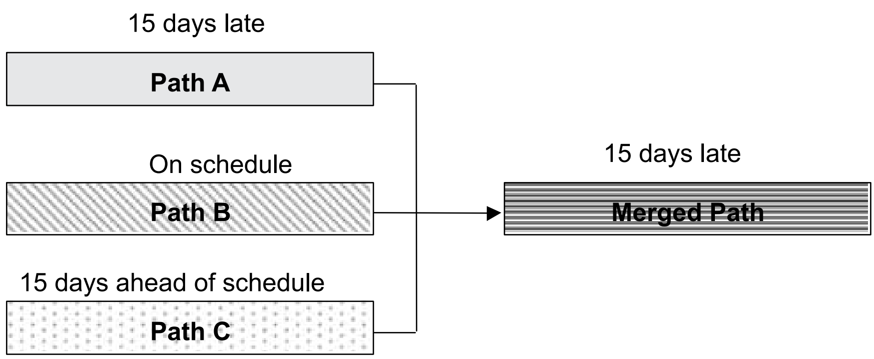


Figure 5.6 Merging paths cause critical chain delay if any of the feeding chains are delayed.

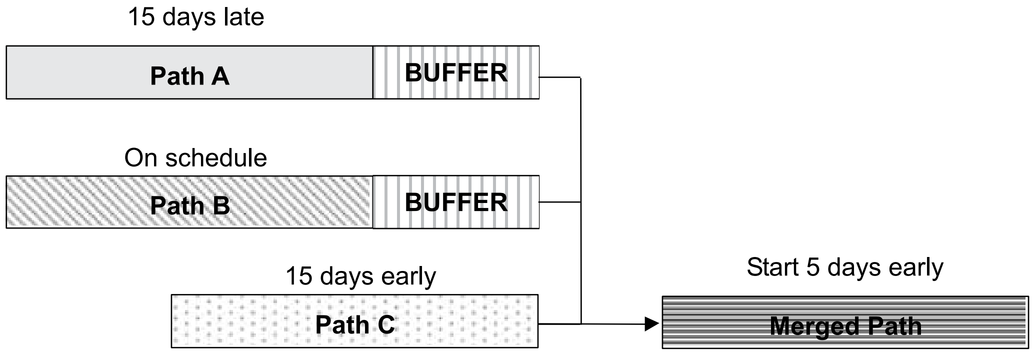


Figure 5.7 Feeding buffers absorb fluctuations in critical chain feeding paths.

while keeping focus on the tasks that need attention to ensure the project completes by the buffered project end date.

Some 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 tasks in the feeding chain. Inserting a properly sized feeding buffer may cause a gap in the critical chain. Because we are not worried about individual task dates that should not be of concern, some Critical Chain software allows removing critical chain time gaps and adding the gap duration to the project buffer. I fear that might reinforce inappropriate attention to the picture of solid task bars versus the understanding of variation. It is not necessary to remove those gaps to support execution.

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 protection for uncertainty relative to other chains. The idea that float or slack can help protect the network from merging chains is fundamentally flawed.

5.2.4 Task Performance

Elevating Date-Driven Performance

Section 4.3.2 described two major factors that lead to wasting of contingency time: student syndrome and Parkinson's law. Individual task completion dates provide an antecedent for these behaviors. Management's response to task performer's behavior to task completion dates reinforces these contingency wasting behaviors. The simple solution is to eliminate task finish dates.

The Chapter 4 discussion on Parkinson's law described analysis of many project's results, revealing 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, which you should have if your schedules do not have buffers and you plan to deliver the project on time, you should have 99% of the tasks reported completed early. Usually, people report most of the tasks as done on the milestone date and they report significant portions of the tasks as late. Eliminating task due dates removes the antecedent for this behavior. Reinforcement of task performance must reward focused task behavior and discourage date-driven behavior.

Student syndrome is wasting task level contingency at the beginning of the task time. 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. They are all of the most important people on whom the Project Manager is counting to get the critical work done. 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. Eliminating task due dates removes the antecedent for this behavior as well.

Elevate Task Performance by Eliminating Multitasking

Chapter 4 described multitasking as the performance of task switching among 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 stomachs and patting their heads at the same time. People actually multitask by dividing time between the multiple tasks. People might do this during the course of the day by working on one task from one project in the morning, and not completing it, and then working on another task from the same or another project in the afternoon, and not completing it either, and then doing the same the next day and so on.

CCPM seeks to eliminate multitasking by eliciting 100% focus on the project task at hand by all resources supporting the project. Thus, eliminating fractional head counts is a primary consideration in scheduling a Critical Chain project.

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 is not multitasking. Multitasking extends the duration of a project task. Unless the manager is also a task resource doing different management jobs in sequence, it does not affect the project duration. As long as you position yourself and your project work to avoid multitasking, you are contributing your best to the project team. In most organizations, people at the working level are not in a position to do that by themselves.

This is where Kanban provides a huge help. It simply is not fair to put the onus on individual workers to decide what task is highest priority for them or to deal with demands for their work by multiple managers. Project Kanban puts in place a system to control WIP at the working group level and removes the onus from the individual workers. The Kanban board is there to control WIP. Managers from elsewhere in the organization are only authorized to override the Kanban board through a defined process that involves the team leader.

5.2.5 Early Start (Just-in-Case) Versus Late Finish (Just-in-Time)

Extensive studies have evaluated the desirability of using early start schedules or late finish schedules. Late finish schedules are an application of just-in-time scheduling of parallel work chains in projects. Project Managers believe early start schedules reduce project risk by getting things done early and late finish schedules:

- 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 noncritical chain (i.e., feeding chain) 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 the task? Does it encourage or discourage the student syndrome?

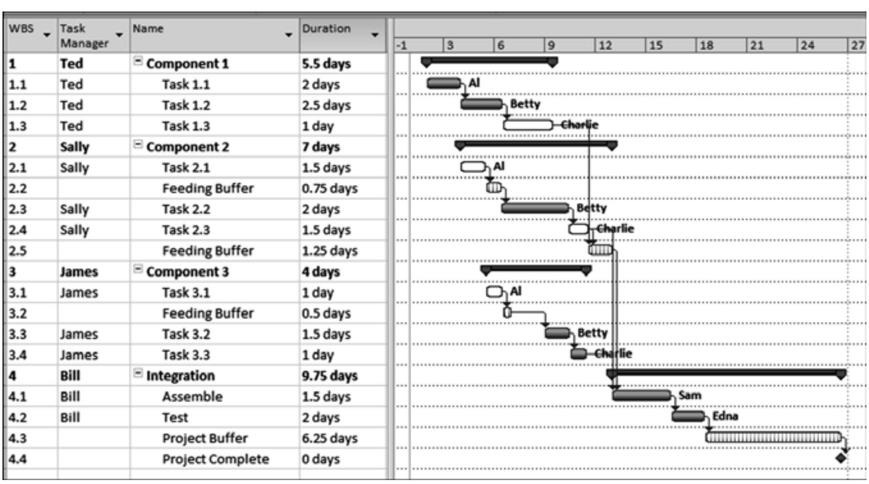
CCPM uses buffered late start for project tasks. The feeding buffers provide an explicitly sized buffer to protect the overall project from late completions in the feeding chains. This maximizes the advantages to the project while ensuring project schedule protection. It reduces project Work in Progress (WIP) compared to early start schedules.

Some projects require early starting of selected tasks for risk mitigation. For example, projects to 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 define 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. There are several ways to do this.

I am often asked, "Yes, but 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.

Figure 5.8 illustrates the Critical Chain schedule for the example project (Figure 5.2) including all of the above features:

1. Task durations reduced to mean durations;



WBS	Task Manager	Name	Duration
1	Ted	Component 1	5.5 days
1.1	Ted	Task 1.1	2 days
1.2	Ted	Task 1.2	2.5 days
1.3	Ted	Task 1.3	1 day
2	Sally	Component 2	7 days
2.1	Sally	Task 2.1	1.5 days
2.2		Feeding Buffer	0.75 days
2.3	Sally	Task 2.2	2 days
2.4	Sally	Task 2.3	1.5 days
2.5		Feeding Buffer	1.25 days
3	James	Component 3	4 days
3.1	James	Task 3.1	1 day
3.2		Feeding Buffer	0.5 days
3.3	James	Task 3.2	1.5 days
3.4	James	Task 3.3	1 day
4	Bill	Integration	9.75 days
4.1	Bill	Assemble	1.5 days
4.2	Bill	Test	2 days
4.3		Project Buffer	6.25 days
4.4		Project Complete	0 days

Figure 5.8 Full Critical Chain schedule for example project.

2. Resource-leveled schedule;
3. Critical chain identified;
4. Project buffer added;
5. Feeding buffers added;
6. Feeding chains "late started" (although it made little difference in this example).

It is ready for execution.

5.3 Exploiting the Schedule Using Buffer Management

Measures drive actions that move you towards the goal. In *The Haystack Syndrome,* Dr. Goldratt noted (Goldratt, 1990):

> 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 5.9 illustrates the cybernetic view of measures used by Dr. Joseph 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*, Dr. Goldratt defined data as, "every string of characters that describes something, anything, about our reality." He defines information as "the answer to the question asked." Dr. Goldratt suggests that the information system should incorporate the decision.

The improved measurement system for Critical Chain Project Management (CCPM) follows the practice established by Dr. Goldratt for production operations. It uses buffers (that is, time) to measure task chain performance. You size

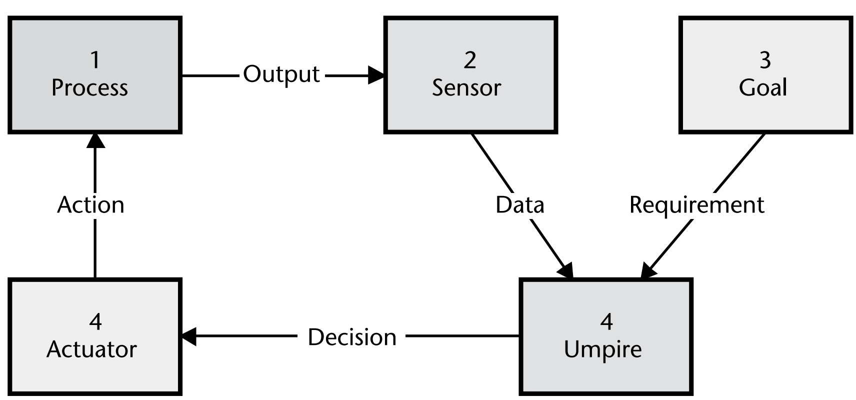


Figure 5.9 Dr. Joseph Juran depicts measurement as part of a control system.

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 feeding buffer. CCPM sets explicit action levels for decisions. The decision levels are based on the rate of buffer consumption measured as the percentage of buffer vs. the percentage of critical chain completed. Figure 5.10 illustrates a fever chart: the preferred tool for presenting buffer use. The fever chart presents the buffer used versus progress along the project critical chain. Decision criteria lines divide the chart into three colored regions. Buffer use versus critical chain progress points are plotted at equal time intervals. The regions drive project team actions:

1. Within the green region of the fever chart: no action;
2. Penetrate the yellow region of the buffer: assess the problem and plan for action;
3. Within the red region of the fever chart: initiate action.

These measures apply to both the project buffer and the feeding buffers. Figure 5.10 also shows an example of how to determine buffer penetration and plot a point on the fever chart.

Statusing project tasks enables comparing the current working schedule to the baseline CCPM schedule to evaluate the effect tasks are having on the project buffer. The current working task on the critical chain maps directly to project buffer impact as shown on Figure 5.10. Tasks on feeding chains also have an impact on the project buffer. However, as they have feeding buffers connecting them to the critical chain, one must subtract the remaining feeding buffer to determine their

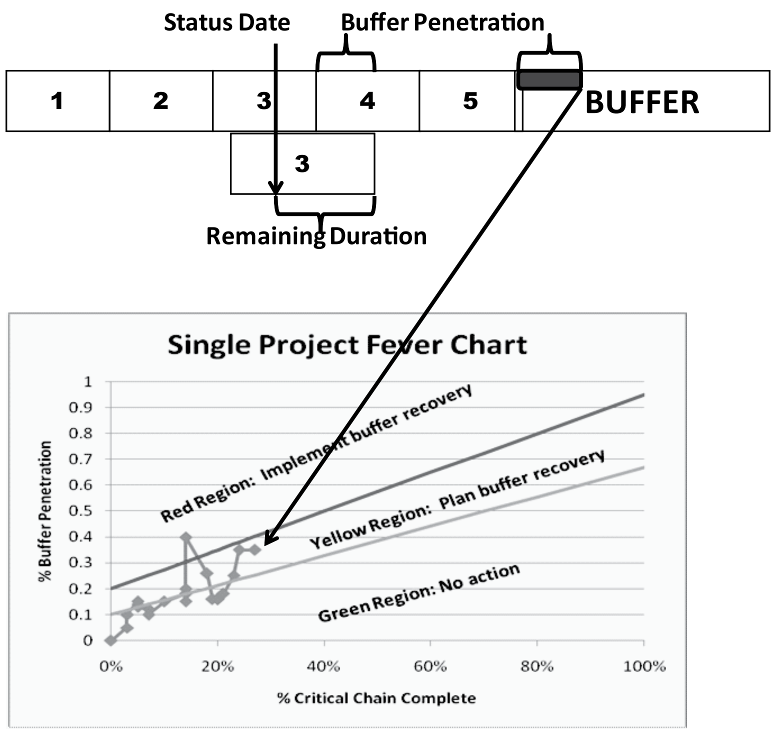


Figure 5.10 Buffer penetration provides action decisions.

impact on the project buffer. Once feeding chains have used up all of the feeding buffers, then they directly impact the project buffer. Software enables easily determining the impact of all tasks in the project on the project buffer. The ones causing the most project buffer penetration may require attention.

Project teams monitor the effect that tasks are having on the project buffer at the appropriate time intervals for the project: usually weekly but sometimes more frequently. For the tool to be fully useful, buffer monitoring must be at least as frequent as one-third of the total buffer time, but usually is much more frequent. I recommend weekly at least.

If the project buffer use is negative (i.e., latest task on the chain is early relative to schedule date) or less than the decision criteria, you do not need to take action: the variation is most likely common-cause. If buffer penetration uses the buffer more than the lower decision line (green to yellow transition) but less than the upper one (yellow to red transition), 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 more than the action line (yellow to red transition), the project team should take the planned action. Through this mechanism, buffer management provides a unique anticipatory project management tool with clear decision criteria. The next chapter addresses in more detail the reason that the decision lines are sloped and logic for setting the end points, but for now the major point is to prevent the early tasks in the project from using up the buffer.

Project schedulers update the buffers as often as they need by inputting actual start and finish dates for tasks into a working schedule and asking for each of the tasks in work at the fever chart update interval (usually once a week[1]) how many days they estimate to the completion of tasks that have been started but not finished as of the status time. This status estimate is called *remaining duration*. They do this without pressure or comment on their estimate. Managers expect these estimates to vary from day to day and roughly half of the tasks to exceed the original duration estimates. As long as the resources are working on the tasks with the CCPM task performance paradigm, managers evaluate them positively regardless of the actual duration.

Updating the buffers requires that you maintain project status versus your schedule in terms of the tasks complete. This is also a useful direct measure of project performance.

Project status data input occurs before project meetings. Project meetings require the statused project schedule with the calculated effect all upcoming tasks are currently having on the buffer to provide buffer management information: when and where to take action to recover buffer.

5.4 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

1. Appropriate timing for fever chart updates depends on the project duration. Weekly is appropriate for projects of over one month total duration. Shorter projects may require more frequent update.

system requirements. Following Juran, Figure 5.11 presents 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. It helped to identify the following set of PMBOK Guide features as the primary ones necessary to deliver to the requirements given in Table 5.1. The correlation matrix also leads to clarification of the requirements that pertain to each feature and therefore supports developing the feature.

The following features, which are (mostly) contained in and explained in the PMBOK Guide, are necessary to satisfy the requirements.

5.4.1 Project Charter

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

Factors That Determine Project Success	**CCPM Features**						
	Project WBS	**Resource Leveled Schedule**	**Critical Chain Identified**	**Mean Task Duration Estimates**	**Project Buffer**	**Feeding Buffer**	**Prioritized Task List**
1.1 Define the project system goal…	●	●	●	●	●	●	●
1.2 Define the project system boundary…	●	●					
1.3.1 Account for common-cause variation…	●		∇		●	●	●
1.3.2 Provide…special-cause variation…	∇		∇				
1.4.1 Identify the project constraint…	●	●	∇				
1.4.2 Exploit the project constraint…	●		∇				
1.5.1 Align project system…psychological…	●		∇		∇		●
1.5.2 Align individual rewards…	●						∇
1.6.1 Define and standardize process	●	∇	∇	∇	∇	∇	∇
1.6.2 Measure process performance.	●				●	∇	∇
1.6.3 Improve processes.	●				∇		∇

● Significant Influence

∇ Some Influence

Figure 5.11 Quality function deployment matrix showing the relationship of CCPM features to requirements.

5.4.2 Project Work Plan

The Project 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 Plan are described in the following sections.

5.4.3 Work Breakdown Structure

The work breakdown structure (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.

I have found some proponents of Critical Chain malign the WBS. My opinion it is that they must not understand how to use it effectively. The WBS has been abused in some organizations by a cost-world focus, but that is not a reason to reject its powerful benefits.

5.4.4 Responsibility Assignment

Responsibility assignment designates individuals responsible to accomplish deliverables in 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 to a named individual while resources assigned to perform tasks can be generic (e.g., pipefitter or programmer). You should also assign unambiguous responsibility to a Task Manager for each task. The Task Manager is accountable to status the schedule for that task. The Task Manager may be a resource working on the task or may be a resource or work package manager.

Milestone Sequencing

Milestone sequencing is a tool to go from the hierarchically formatted WBS to a logical project schedule. It is most useful on larger projects, that is, more than a few hundred activities and larger than a few million dollars. 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.)

Some organizations specify a number of stage gate milestones projects must go through. They can provide the backbone for a milestone sequence chart.

Milestones are often associated directly with dates. Milestones need not have fixed dates associated with them. They can represent a key technical accomplishment and sometimes serve to simplify project network building. With CCPM all dates that are associated with a schedule must have a buffer. So if you need to associate dates with some milestones, be sure to buffer those dates. Milestone buffers should not directly affect the project flow. Milestone buffers should precede the milestone date in the schedule and provide a fixed date like a project buffer. Due to your scheduling tool and practice the Gantt chart might show a milestone symbol before the buffer but it should not show a date.

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 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, resource requirements, and the assumptions necessary to support these estimates.

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 task 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.

5.4.5 Project Quality 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 identified 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. Ireland (1991) provided an overview of a satisfactory process to meet these requirements.

5.4.6 Project Change Control

The project measurement and control process will, from time to time, trigger the need for action to complete the project successfully. Additionally, unfulfilled assumptions made at the start of the project, for example, 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 all of the project team.

5.4.7 Project Risk Management

Project risk management handles potential causes of special-cause variation. Because 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.

5.4.8 Project Kanban

The function matrix also includes Project Kanban as the feature to control WIP at the working level and handle the potential intrusion of nonproject work into the project delivery process. Project Kanban reinforces the necessary CCPM behaviors such as focusing on one task at a time, statusing the CCPM schedule, and buffer management. Chapter 3 provided the fundamentals of project applications of Kanban. Implementation is straightforward and can be done independently of project scheduling. The tie to CCPM is that for each work group deploying Kanban a task in the CCPM schedule carries a Kanban card and will be managed with the team's rules for Kanban. This approach puts an immediate limit on WIP at the working level and greatly aids reducing multitasking. However, if the management team has not yet bought into the idea of limiting WIP, there can be initial barriers for the teams as they will continue to get pressure to multitask.

If the team has a repeatable process for the work they do, the Kanban board can also simplify schedule statusing. In most applications the Kanban cards have spaces for actual start and finish dates. The columns on the Kanban board can be associated with remaining days of schedule duration based on the board history. Thus, a person can read the team's status on all project tasks directly from the Kanban board and input it to the scheduling program for buffer calculations. The results of buffer calculations can affect the priority of tasks working and in the backlog.

There are several electronic tools available for Kanban. One of my favorites is LeanKit Kanban. You can try it for personal use at LeanKitKanban.com, but I must raise a huge caution. Doing Kanban only electronically does not work as well as a local always-visible board because it does not provide the focus reinforcement of a visual control in the workplace. While the electronic tools can be highly valuable, particularly for geographically dispersed teams and for data and information generation and analysis, the Kanban board must be visible in the workplace full-time to serve as visual control and thereby reinforce the desired CCPM behaviors. Some organizations purchase large-screen TVs and use an old computer to display the board. That works well, but a simple board of Post-it® notes on a paper background is far superior to an electronic only method without the full time visual display.

The primary team behavior that makes Kanban work is to have a brief standup meeting each morning at the Kanban board. During that meeting, people can move cards as necessary although there is no reason to restrict them from moving cards before the meeting. The purpose of the meeting is to ensure action on anything that is blocking progress on the tasks that are in work. The board manager's most important job is to help make that happen. In most cases the board manager is also the Task Manager for the tasks on that board, the first-level supervisor of the work group.

The board manager also must have a scheduled (I recommend weekly) meeting with the board customers to pick the tasks that will be promoted to the input queue on the board. The process must also allow for exceptions to handle emergencies and cases where the queue is threatening to run dry before the next scheduled meeting.

The input queue has a WIP limit, so in the beginning, these meetings can be contentious. Once people learn the value of limited WIP to accelerating the completion of all work, that contention moderates. These meetings decide the relative priority of project work and nonproject work. It helps to have rules for this such as "red buffer project tasks normally take priority over all other work." The process must allow for formal exceptions to any such rule to handle unforeseen emergencies or other cases.

Once a week the board manager should also meet with the team to determine how items in their work processes, including how the board is used, can be improved. Improvement of those processes can use standard Six Sigma or Lean approaches.

Finally, once a month, the Senior Management team should meet with all of the board managers to plan improvements to the overall processes for CCPM and Kanban. Selected improvement projects at this level can also deploy Six Sigma and Lean improvement methodologies.

5.5 Summary

This chapter developed the project system requirements and described the single-project Critical Chain features and key supporting PMBOK Guide system features designed to satisfy those requirements. The 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 durations and a project buffer.
- Feeding chains and resource efficiency are subordinated to the critical chain with 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.
- Project Kanban controls WIP and integrates project and nonproject work.
- Continuous improvement of all of the processes needs to be part of the system.

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

References

Goldratt, Eliyahu M. (1990). *The Haystack Syndrome*. Croton-on Hudson, New York: North River Press.

Ireland, L. (1991). *Quality Management for Projects and Programs*. Upper Darby, PA: PMI.

Juran, J. (1988). *Juran on Planning for Quality*. New York: The Free Press.

Kiley, Martin D., (1996). *1997 National Construction Estimator*. Carlsbad, CA: Craftsman Book Company.

Meredith, J., and Mantel, S. (1995). *Project Management: A Managerial Approach*. New York: John Wiley & Sons.

Moore, David S., and McCabe, G. (1993). *Introduction to the Practice of Statistics*. New York: W. H. Freeman & Co.

Popper, K. (1972). *Objective Knowledge: An Evolutionary Approach*. Oxford: Clarendon Press.

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

CHAPTER 6

Starting a New Project

6.1 Project-Initiation Process

The project-initiation process ensures meeting all of the conditions necessary for project success.[1] 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 6.1 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 Plan that is sufficient to start work on the project.

The PMBOK Guide (PMI 2013) separates the project-initiation process, which includes identifying stakeholders and creating the project charter, 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.[2]

6.2 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 (1996) as essential:

- Vision;
- Purpose;

1. This text assumes a set of approved projects. For my take on project portfolio selecting applying TOC thinking to portfolio management (including project selection), see Leach (2005).
2. Some in the TOC community refer to "full kitting" of a project before initiating it. Apparently this term stems from production terminology and includes many of the ideas presented in this chapter. I prefer to retain the well-established project management terminology and content. I fully support the idea.

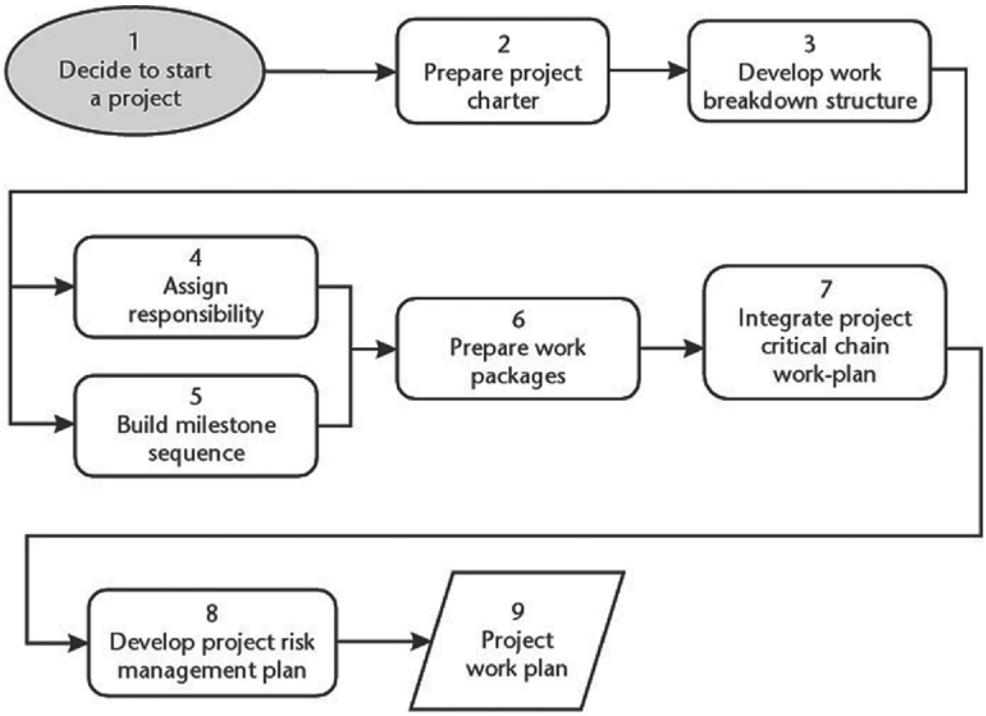


Figure 6.1 The project-initiation process.

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

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

6.3 Stakeholder Endorsement

The PMBOK Guide (2013) defined project stakeholders as (p. 563): "An individual, group, or organization who may affect or be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project."

One of the major changes to the fifth edition of the PMBOK Guide was to add project stakeholder management as a tenth knowledge area addressing:

- Identify stakeholders;
- Plan stakeholder management;

- Manage stakeholder engagement;
- Control stakeholder engagement.

For most projects the stakeholder 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. You need to assure that your team has listened to and addressed the needs of each stakeholder. 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.

6.4 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 find it difficult to develop 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 and we will address that in Section 6.7.2. The WBS is your tool to organize the entire scope of the project and to assign responsibility to produce everything.

6.4.1 TOC Approach to Project Schedule Network Building

TOC advocates offer two approaches to develop the project schedule network. Most TOC advocates bypass the WBS. One approach, the prerequisite tree (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* (Goldratt, 1994) is not appropriate for generating a project WBS. The reason is that the PRT 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. Dettmer (2010) described a modified method to ensure both necessity and sufficiency.

A second approach, adopted by many in the TOC community, uses 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 potential 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 build the schedule network 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.

Nonetheless 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. You can also use the backwards approach to check your network asking if each task has the predecessors necessary to provide the needed inputs. However, backwards planning alone does not substitute for an effective WBS, especially on larger projects.

6.4.2 The Conventional WBS

Several useful standards may aid you in developing your WBS. The PMI provides a good one, their *Practice Standard for Work Breakdown Structures* (PMI, 2006) and the U.S. DOD has a very comprehensive one (DOD, 2004). Perhaps PMI's Practice Standard will help some people in their development. In addition to providing many examples, it outlines the quality characteristics a WBS should exhibit in terms of core and use-related quality characteristics. They include:

- Defines scope of project;
- Organizes deliverable-oriented grouping;
- Arranges all major and minor deliverables in a hierarchical order;

- Employs a coding scheme for each element that clearly identifies where it resides in the hierarchical order;
- Contains 100% of the work defined by the scope;
- Contains elements that are defined using nouns and adjectives—not verbs
- Enables assignment of accountability to the appropriate level;
- Is broken down to a level sufficient for managing the work.

The PMI practice standard also includes a diagnostic checklist intended to aid identifying when there are problems with the WBS (pp. 24 and 25).

Harold Kerzner provides the following criteria for a WBS (Kerzner, 1992):

> [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 versus the work package comprising an aggregation of a number of tasks to produce a major deliverable that I describe. Some also use a WBS dictionary. A WBS dictionary should make sure that the deliverable is unambiguously defined so it minimizes the possibility of miscommunication. Others substitute a scope statement or statement of work (SOW) for the same purpose.

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 product or service. 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. As 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 or checklists 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. Some use a list of questions to stimulate the group's thinking, for example:

1. Are there special environmental or safety issues on this project?
2. Are all licenses, permits, and regulatory requirements covered?
3. Are there any special tools that need to be developed?
4. Are there special software deliverables in addition to those already included in the WBS?
5. Are there studies (interchangeability, uprate) or other investigations that have the potential to result in increased scope not included in the WBS?
6. Are there any nonstandard contract requirements?
7. Are there auxiliaries deliverables that are not included in the WBS?
8. Are there customer options that can be specified or exercised later that we should provide for 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.

For multiple projects the project designator can be used as a prefix to ensure unique WBS numbers for every task. This is important when people and groups perform work for multiple projects. A similar system to provide unique identifiers to nonproject tasks provides the project WBS benefits to nonproject work.

6.4.3 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. Because 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 have the flexibility and authority to design their WBSs, select their project teams, and assign responsibility and authority. If they do not have that authority, I consider them more of a project reporter.

Many organizations have adopted the idea of a Project Management Office (PMO). Although the organizational strength of PMOs varies, in most cases the PMO sets the standards for Project Managers to follow in regards to Project Plans and scheduling. That is particularly useful for organizations that perform multiple simultaneous projects.

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. As it is unlikely that the WBS was organized this way, the project-management team may cross-cut 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. These connections are the most vulnerable part of project flow and must be understood to ensure work synchronization. For projects the scheduling software provides the predecessor and successor links for tasks. You should provide a similar field on your Kanban cards for nonproject work that requires input from other work groups or provides outputs that go to another work group.

6.5 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 6.1 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.

An enhanced form of responsibility assignment that has found wide use is a RACI format. R is for responsibility, A is for accountability, C is for communicated to, and I is for informed. The idea is to indicate for each work package in the WBS who fits into each of those categories. There should be only one person, by name, in the accountability category. The other categories can be by organization designation and include multiple parties. RACI's use of responsibility may differ a bit from your definition. In a RACI matrix it means the resources that actually do the work. The accountable person would usually be the Task Manager at the work package level of the WBS and more Senior Managers above that level.

6.6 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 schedule must logically sequence all of the project tasks. For a small project (i.e., 50 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.

An effective way to aid developing the logic is first to identify the major project phases in terms of key milestones. Figure 6.2 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

Table 6.1 Example of a Linear Responsibility Matrix WBS Number Deliverable Responsible Person Notes

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	

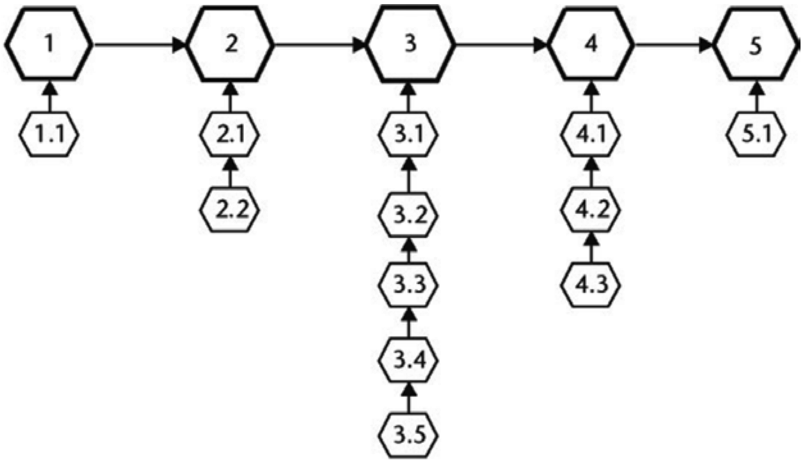


Figure 6.2 The key milestones define a backbone for the project-task sequences.

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 6.7). 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 schedule, 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 6.3 illustrates this idea. You should still control the project to the overall project buffer.

6.7 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 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

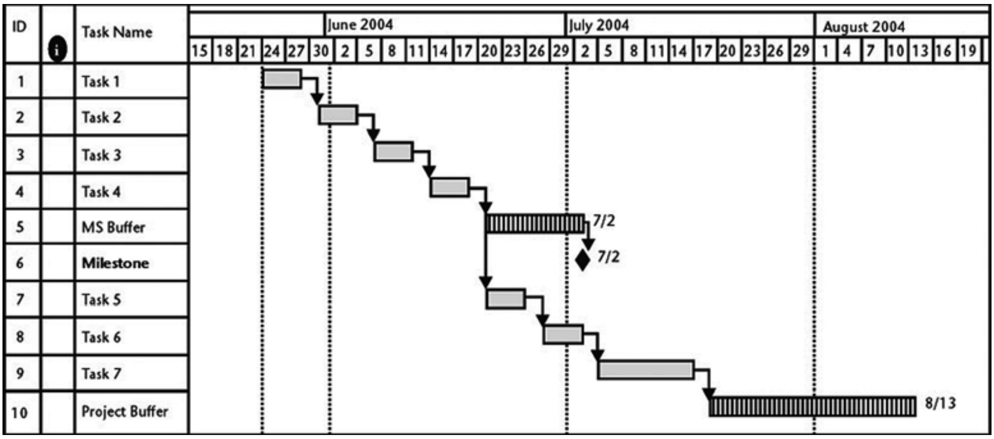


Figure 6.3 If your organization uses milestone dates to judge project progress, you must put a buffer in front of them.

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 6.4 illustrates the project-logic input, an essential part of the work package that, combined with the assumptions and deliverables (scope statement), provides the information necessary for a project schedule.

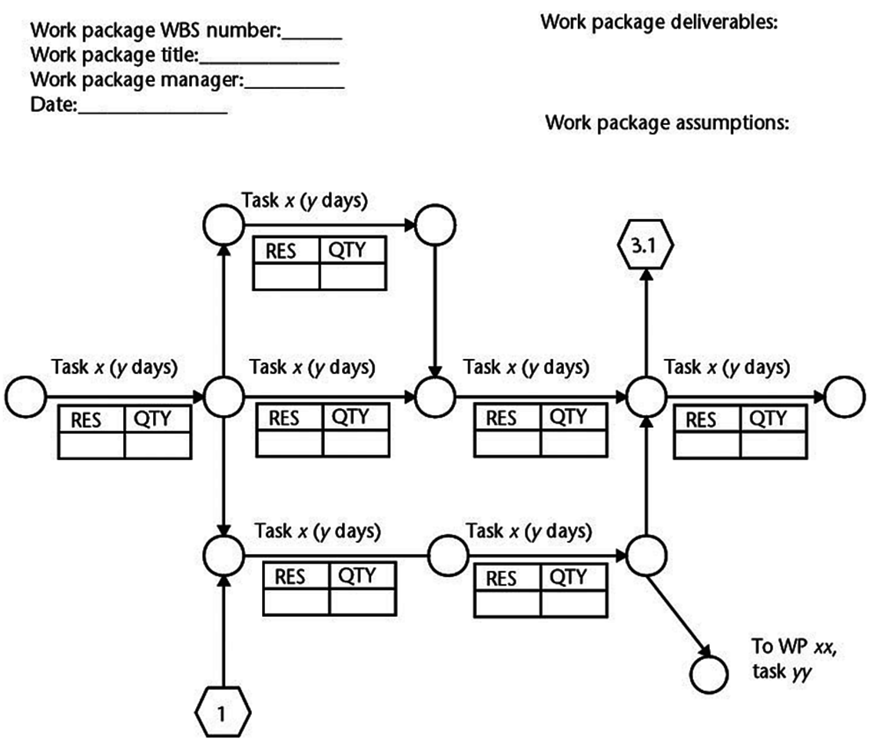


Figure 6.4 The work-package logic provides essential input to create the project schedule.

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.

6.7.1 What Comprises a Project?

An initial decision you might need to clarify for your organization is when to define some piece of work as a project and manage it using the CCPM methods. You might need to define both a minimum and maximum size for a project. The minimum size specification needs to define roughly what compromises a large enough chunk of work to justify the planning and control provided by project management. Your definition might include characteristics such as, "Work that requires resources from more than one work group." That is a sensible definition if you are using a tool such as Project Kanban to control work at the work group level. Work that can be performed totally within one work group can be satisfactorily controlled using that work group's Kanban process. Some organizations use the estimated total amount of work for the task, for example, over 200 work hours, to define what should be treated as a project.

You also should define a maximum size for a project. I prefer to define it in terms of time: projects should not exceed one year in duration, including the project buffer. The reason is that effective control of project work requires a buffer size that is sufficiently sensitive to effect timely buffer management (i.e., on a weekly basis). A three-month project buffer broken into three regions means that each region is nominally a month long. Allowing work to fall behind a month before taking action is usually not acceptable. A one-year total project duration means that the project work schedule should not exceed about eight months of work. If your project extends beyond that time frame, break it into a number of overlapping schedules and treat them as a program.

6.7.2 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.

Sometimes assumptions also relate to the result of the project because the project is to create the design for that product and the planning for completion of the project needs some idea of those design parameters. Those assumptions must be clearly stated as well.

You may need to work 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 are 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."

6.7.3 Project Schedule Network

The most important part of the work-package documentation is the project schedule network input. Note that the work package network input does not carry dates. It provides activity duration and logic and specifies resource requirements for the activities. One reason that 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. Recall from your understanding of variation that 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 schedule software will calculate early and late start and finish dates for each task. You should completely ignore those task 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. They talk in terms of accounts such as cost accounts and summary cost accounts instead of in terms of project deliverables. Such terminology puts the emphasis in the wrong place. Planners have to develop the schedule separately and figure out how to make the schedule match the budget thinking. 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. The reason is that with CCPM 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.

What Comprises a Project Task?

PMI's *Practice Standard for Scheduling Second Edition* (2006-2) defined a task as, "A term for work whose meaning and placement within a structured plan for project work varies by the application area, industry, and brand of project management software." I do not find that definition useful and have no idea why they think the brand of project management software has something to do with defining a task. They also define activity as: "A component of work performed during the course of a project." I like that better, but I define task and activity as interchangeable terms for the lowest level of the WBS, a value-adding packet of work that forms the basic building block of a schedule or network. An activity is performed by one group of resources. With that definition, it may help you to think of tasks in project schedules as a road map of the handoffs of work from one work group to another.

While I included the task as the lowest level in the WBS, formally the WBS ends with the work packages. Although the WBS numbering flows down to the tasks, they usually are not considered part of the WBS.

Project tasks should be titled with at least two words: a verb and a noun. Focus on a task as a value-adding step in the project. It does something (the verb) to something (the noun). It has necessary inputs and creates at least one output. A project task must have an estimated duration and identify the resources necessary to perform the task. Project tasks must also have at least one relationship to other tasks in the project schedule network.

Project task names should be uniquely identifiable in all forms they are used. For example, you should not have 50 tasks all titled "create drawing." While one might be able to infer the drawing of what by the WBS number (a reason to always keep the project name and WBS number associated with a task), it is much more helpful to have the task name itself provide at least broad identification (e.g., "Create electrical drawings for control module").

Projects often require ongoing work that does not belong in the project schedule. These are often called level of effort (LOE) work. LOE work includes such things as project management and administration and quality and safety oversight. Such elements need to be included in the project budget but do not belong in a project schedule. They do not directly affect the flow of value adding work in the project and interfere with the network calculations.

Project Schedule Network Logic

The project schedule network 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 because 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 that 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 schedule. Link the subtasks underneath the summary tasks. Chapter 7 describes how to create the Critical Chain schedule from this resource-loaded project network.

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 who 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.

It is acceptable to use a designator for a group with a Kanban board as a resource. In that case you might input the WIP limit for the Kanban board as the quantity of available resources for the purpose of controlling demand in the project schedule (resource leveling). If the work group also performs nonproject work, you might need to adjust for that.

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 schedule moves a task to level resource demand can depend on the task length and location in the schedule as well as the resource loading.)

Figure 6.5 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

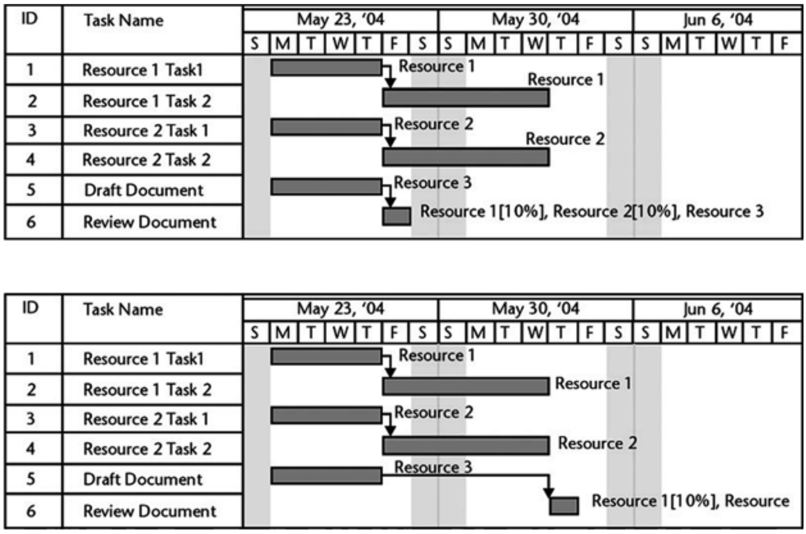


Figure 6.5 Example of resource leveling causing an unexpected result: (a) before leveling; and (b) after leveling: the document review gets displaced to the right due to 10% loading of resource.

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 schedules.

How Many Tasks (Also Known as Granularity)?

Many project schedules 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 schedule and the TOC leads to the recommendation that project schedules be only as detailed as you need to run your project successfully and no more. Project tasks generally should represent a handoff of a specific artifact from one resource (or group of resources) to another.

Project schedules 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 anything other than schedule management. Your highest-level project schedules should only rarely exceed a few hundred activities. Larger projects require a hierarchy of schedules where no individual schedule contains more than a few hundred activities. In these cases, you should link from lower-level schedules to the higher-level schedules at the start of the lower-level schedule's project buffer.

The primary reason to limit the number of tasks in a project schedule is that overall uncertainty does not justify too much detail. Too much detail increases the work to create and maintain the schedule, as well as the probability of errors in the schedule. Very few people can understand a schedule with more than a hundred activities, even with study. As the number of potential links in a schedule increase exponentially with the number of tasks, it is highly unlikely that a schedule with even 100 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 100 activities is 1% of the total project. As most of the tasks can be estimated with accuracy no better than ±50% to several hundred percentages, it makes little sense to divide the project into smaller pieces.

People often suggest that an insufficiently detailed schedule is a cause of project failure. When projects fail, they usually do so in the range of several hundreds of percentages (i.e., the cost two or three times the initial estimate) not fractions of 1%. It is not logical to conclude that schedules 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 schedule. A given planning effort spread over more tasks means less effort per task. This may lead to a less accurate schedule, not a more accurate one. 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 schedule.

For statistical reasons, there is value in ensuring that the critical chain of your schedule 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.

However, 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 less important on feeding chains because the schedule protects feeding chains with both the feeding buffer and the project buffer.

6.7.4 Activity Duration Estimate

Chapter 4 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 are based on 100% effort on the task for the resource that determines the task duration. If task durations are based on fractional resource loading, first reduce the task duration keeping the work (person-hours or person-days) the same. For example, if the task has an engineer at half time for 10 days, first 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 schedule 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. The method for sizing the buffer suggested by Goldratt is to size the total schedule buffer as one-half of the task time of the chain it is protecting. A slight modification of that buffer sizing method is to size the buffer as one-half the difference between a low-risk duration estimate and the mean duration estimates summed along the chain. This allows for relatively long fixed duration tasks (e.g., material lead times) to not contribute to the buffer. When using this method, the low-risk duration estimates are simply twice the mean duration estimates for all but the exceptional fixed-duration tasks.

The above method for task duration estimating assumes that the original estimate was based on the resource that determines that the task duration is focused 100% on only that task. CCPM implementations quickly showed such was not the case: even the critical people were often fractionally assigned to tasks, sometimes as low as 20% assigned. In those cases, one should first divide the usual task estimate by the fraction (or multiply by the percentage as a decimal) before cutting it to account for variation.

Although more sophisticated methods have been proposed (including by me in earlier versions of this book), practice has shown the above method of task and buffer sizing method fine in all cases. I will later offer a method to revise your buffer sizing rule should your data justify doing so, but so far all of my successful clients have stayed with the simple 50% rule for buffer sizing.

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 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., one-half) to generate the project buffer. Including an allowance for variation in individual low-risk task duration estimates acts as an antecedent to undesired behaviors such as Parkinson's law and student syndrome, or other task duration or date anchors, thereby needlessly extending the project schedule and causing performance to be less than achievable. All allowances for duration variation should be accumulated into buffers at the end of chains of tasks.

6.7.5 Uncertainty Revisited

Project Managers face a conflict over the uncertainty of estimates. Pressure comes from management or clients who might 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.

As 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 it is. The second-most important question is how big it is. 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? It is the 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) said, “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 10, alas.) After a few intermediate steps, the text stated, “The definitive estimate, also referred to as detailed estimate, has an accuracy of ±5%.” We just agreed above that the actual cost of a very well-known, existing automobile (on the lot, so to speak), that is, identical items, at the same time and place, can vary by twice that amount. How could we possibly expect an estimate of a lesser-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* (Vigder and Kark, 1994) stated:

> 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.

I could find no project management books providing an operational definition of the term *accuracy* as, for example, ±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 you be sure that your project stakeholders 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 2 if you need some reminders) and perhaps even of some projects that spent the project's entire initial budget 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? Elsewhere I describe the well-known human cognitive bias known as the optimism bias.

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 budget and schedule 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 overruns.

If the project behavior is as we hypothesized in Chapter 4, that is, no credit for positive variances, it only takes one late task on the critical path to make a project late. In addition, because 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 as 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 2 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. With a few notable exceptions for tasks that do not have much variation, 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.

6.8 Need for Cost Buffer

If cost is important in your world, your Project 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 that 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" (Leach, 2003) the two kinds of uncertainty for which you are accounting:

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 (1994) performed a recent study on software projects and noted:

> 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 7.5 describes cost buffer sizing.

6.9 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.

6.10 The Project Work Plan

The project work plan,[3] sometimes called the Project Plan (PP), Project Execution Plan (PEP), Project Management Plan (PMP), or something similar, 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 (PMI, 2013) noted that the Project Plan is used to:

- Guide project execution;
- Document project-planning assumptions;

3. Many people refer to the project schedule as a plan. It is in the generic sense, but the Project Plan discussed here is much more than just the schedule. I try to differentiate but sometimes also mistakenly call the schedule the plan.

- 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 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.

6.11 Change Control

Change control 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.

6.12 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.

6.13 Summary

This chapter provided the process and tools necessary to create an effective project work plan. The key elements that I have identified as 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.
- Single project durations, including buffer, should not exceed one year.
- 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 that you put into the Project Plan and degree of formality that 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

CH2MHILL. (1996). *Project Delivery System: A System and Process for Benchmark Performance.* Denver, CO: CH2MHILL.

Dettmer, H. William. (2010). *The Logical Thinking Process*. Milwaukee, WI: ASQC.

DOD. (2004). U.S. *Department of Defense, DoD Handbook—Work Breakdown Structures*. MIL-Goldratt, Eliyahu, M. (1994). *It's Not Luck*. Great Barrington, MA: North River Press. HDBK-881, 1998. http://dcarc.pae.osd.mil/881handbook/milhdbk. Accessed May 24, 2004.

Kerzner, Harold. (1992). *Project Management, A Systems Approach to Planning, Scheduling, and Controlling,* 4th ed. New York: Van Nostrand Reinhold.

Kiley, Martin D., and Marques Allyn. (1997). *1997 National Construction Estimator*. Carlsbad, CA: Craftsman Book Company.

Leach, L. (June 2003). "Schedule and Cost Buffer Sizing: How to Account for the Bias Between Project Performance and Your Model." *Project Management Journal*, Vol. 34, No. 2357.

Leach, L. (2005). "Applying the Theory of Constraints to Project Portfolio Management." In Levine, *Project Portfolio Management* (pp. 357–390). New York: Jossey-Bass.

PMI. (2006). *Practice Standard for Work Breakdown Structures,* 2nd ed. Newtown Square, PA: Project Management Institute.

PMI. (2006-2). *Practice Standard for Scheduling, Second Edition.* Newtown Square, PA: Project Management Institute.

PMI. (2013). *A Guide to the Project Management Body of Knowledge,* 5th ed. Newtown Square, PA: PMI.

Vigder, M. R., and A. W. Kark. (1994). *Software Cost Estimation and Control.* NRC-CNRC (National Research Council Canada), NRC No. 37166, February 1994.

CHAPTER 7

Developing the (Single-Project) Critical Chain Schedule

This chapter first presents the overall process to create the single-project Critical Chain schedule 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 schedules.

You can think of your schedule as a model of reality. As with all models, it is not reality. It is a greatly simplified model of reality. It needs to be good enough to help you execute the project in a way that produces the project result as soon as possible.

7.1 Process

The basic steps of the process to create a single Critical Chain project schedule follow. If you have a Critical Chain scheduling tool, assuming you have set your scheduling defaults correctly, the process is to:

1. Identify the constraint:
   - Create the schedule network.
   - Resource level the network.
   - Identify the critical chain.
2. Exploit opportunities to shorten the critical chain:
   - Look for opportunities to change or add resources to remove tasks from the critical chain.
   - Look for opportunities to change task relationships to remove tasks from the critical chain.
   - Reduce task duration estimates.
   - Add the project buffer at the end of the critical chain.
3. Subordinate noncritical chain tasks to the critical chain:
   - Add feeding buffers at merge points with the critical chain.
   - Late start feeding chains.
   - Remove resource overloads resulting from the addition of buffers.
4. Elevate the constraint:
   - Add or change resources at strategic locations to shorten the critical chain.

- Revise processes on critical chain tasks to reduce their duration.

5. Avoid inertia:
   - Save the baseline schedule.
   - Return to step one to explore further opportunities to optimize the workflow.

If you are in a multiproject environment, you then need to Pipeline your schedule as discussed in the next chapter. This may entail moving the start date of the project.

The following provides a procedure assuming that you are not using a Critical Chain computer scheduling tool. Project Managers have used the manual approach to successfully plan and run single CCPM projects of up to $5 million. Larger projects, projects in a multiproject environment, and environments that lack scheduling expertise require Critical Chain scheduling software.

Working some networks manually helps you to understand the problem that the computer is solving. The knowledge gained will help you 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 duration estimate (nominal 50–50 time) and primary resource requirements. (For tasks with multiple resources, identify the primary resource you believe will determine the task duration. If there are several resources that seem to determine the task duration, consider breaking the task up to have only one primary resource for each task.)

   1.2. You can use features such as the resource graph in MS Project software to identify where in the schedule resource demand is high. If you do not have resource overload in your project, go to step 1.6.

   1.3. If you do have resource overloads, then identify the resource that you will level to eliminate the overload first. This should be the overload nearest to project completion or the one that shows the most overload. If several show about the same amount of potential overload, then choose the first one you come to while working backwards from the end of the schedule.

   1.4. Remove resource overload by resequencing tasks earlier in time. (Do not worry about creating new overloads for other resources 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 overloads for the next resource, you must avoid creating overload 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 schedule to determine if resequencing can shorten the overall project duration. If so, do it. Do not trial-and-error too many

solutions; you are seeking a good-enough result, not an optimum answer.

   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. You can identify the feeding buffer locations working backwards along the critical chain. For any critical chain task with more than one predecessor (its critical chain predecessor), insert a feeding buffer between the other predecessors and the critical chain successor.
   3.2. Size the feeding 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.)
   3.3. Resolve any resource overloads created by adding feeding buffers through resequencing tasks earlier in time.
   3.4. Move to an earlier time any dependent tasks preceding those moved.
   3.5. Recheck resource leveling.
4. Elevate (shorten) the lead time of the project by using added resources for certain windows of time to break overload.
5. Go back to step 1. Do not allow inertia to become the constraint.

A Critical Chain schedule 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 finish dates. They are meaningless. I used to suggest talking about the Critical Chain plan rather than the Critical Chain schedule but found that confusing relative to the content of the Project Plan. So although I use the word schedule and computer programs assign start and finish dates to all tasks, please keep reminding yourself and others that task dates are meaningless; they all have a probability of zero because of variation. You should concern yourself with the flow of work and not with task dates.

7.2 Good Enough

"Good enough" is an important idea in developing Critical Chain project schedules. For mathematical reasons, it is impossible to build a precise optimizing algorithm for resource leveling. The procedure to develop the Critical Chain schedule ensures that the schedule that you build will be good enough. This means that the overall length of the schedule will be, within a small fraction of the length of the project buffer, nearly the shortest or optimum schedule path. Because reality will change many assumptions and we cannot explicitly predict the results of statistical fluctuations, this is good enough. It is good enough if it causes people to work on one task at a time until it completes (relay racer task behavior). This encourages Rule 1: Focus. It is definitely better if the focus is on the right task at any given time, but even if tasks are done in the wrong sequence from a system throughput viewpoint, overall throughput will still improve by reducing multitasking and date-driven task behaviors.

7.3 Examples and Practice

7.3.1 Small Example

Figure 7.1 illustrates a small example without resource overload to work into a critical chain. The figure illustrates the schedule 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 parentheses is the task duration in days. Note that task 3 depends on completing tasks 1.2 and 2.2.

Following the procedure, 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 7.2).

Next, identify the critical chain and add the project and feeding buffers (see Figure 7.3).

Now consider the same little project with resource overload. Figure 7.4 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., red = engineers, blue = musicians or equipment operators).

Lay the network out with all of the tasks pushed as late as possible. In this case, all you have to do is add a "start as late as possible" constraint to task 2.1 (Figure 7.5).

Next, remove the resource conflict, working backwards from the end of the project (Figure 7.6).

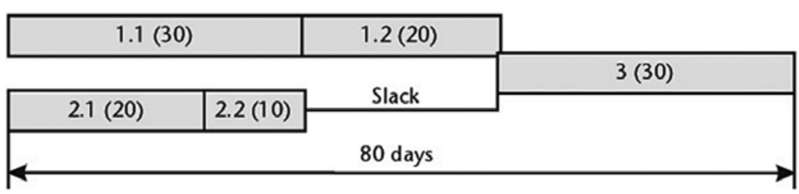


Figure 7.1 A simple project illustrates a normal early-start schedule.

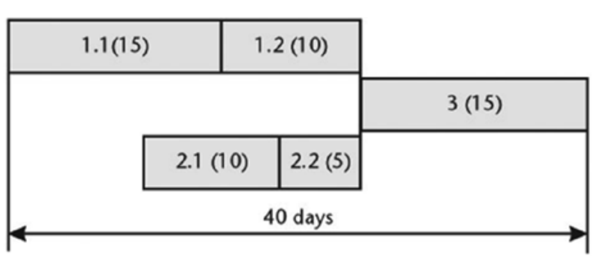


Figure 7.2 The first step to create the critical chain reduces the task times and organizes tasks to a late-finish schedule.

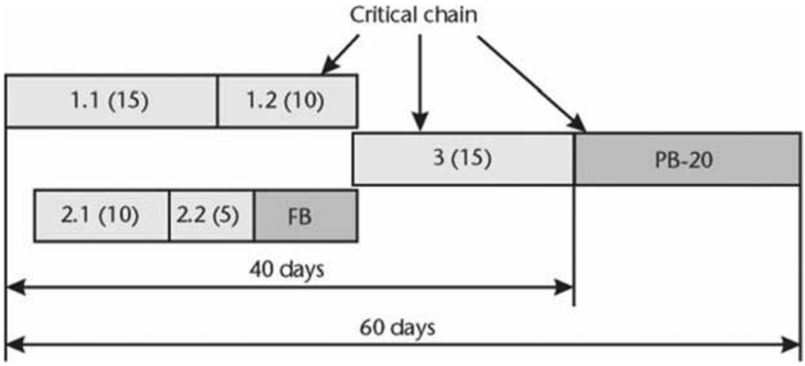


Figure 7.3 With no resource overload, just add buffers.

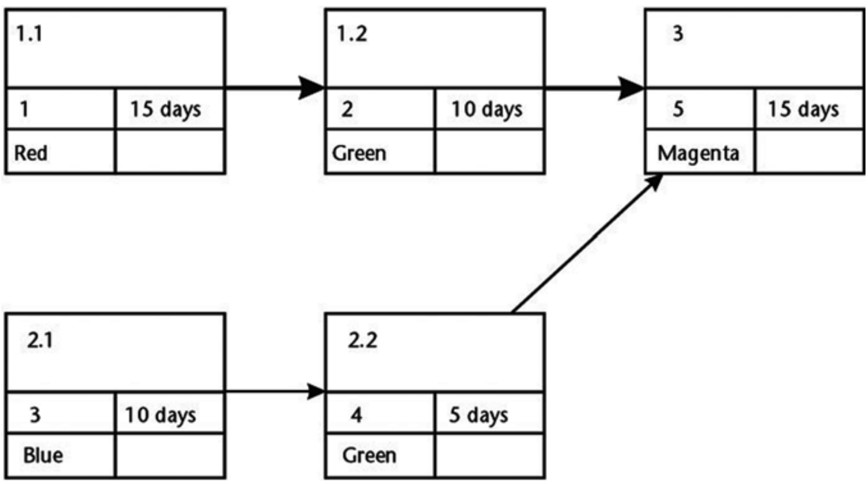


Figure 7.4 The same projects with specific resource assignments.

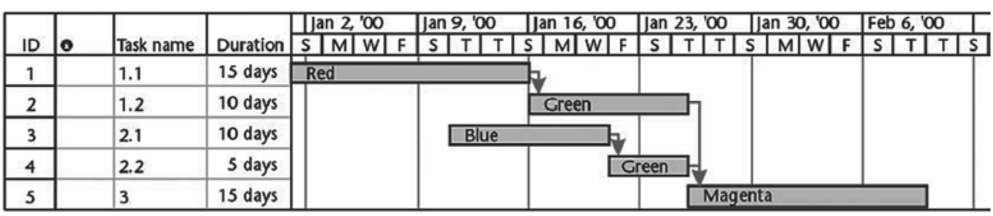


Figure 7.5 The first step pushes tasks to the late finish.

Figure 7.6 illustrates the two ways that we could resolve the green resource overload. Note that each schedule shows a new dependency for the resource constraint.

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 you can choose either one as the critical chain. Add the project buffer and critical chain feeding buffer to each option and see what happens.

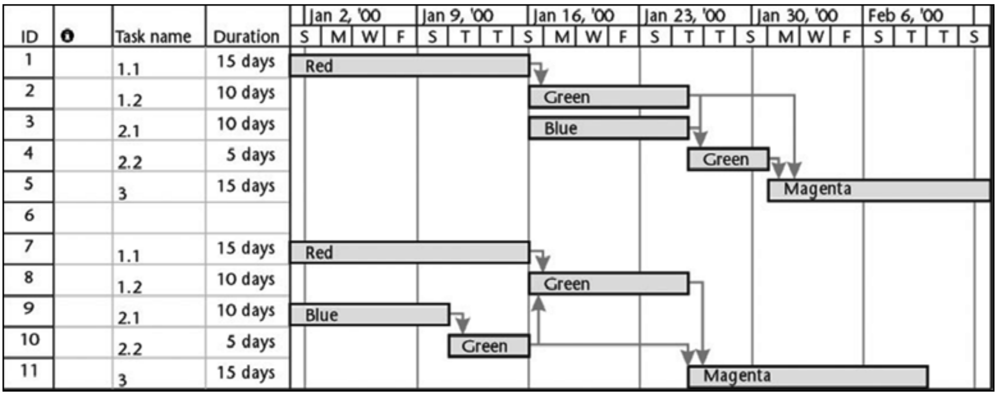


Figure 7.6 Alternative ways to resolve resource conflict.

Figure 7.7 illustrates the Critical Chain schedule for the upper choice of Figure 7.6 for removing conflict for the green resource. The critical chain comprises all of the project tasks 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; you just have the extra lead time for the noncritical 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 overload (Figure 7.8).

Figures 7.7 and 7.8 illustrate three different, but completely valid, Critical Chain schedules for the project presented in Figures 7.4 and 7.5. Any one of those schedules is suitable for the schedule because of the small differences compared to the project buffer. It often works out a little better to resolve overload 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 7.9 exercise as a Critical Chain schedule 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 7.10 for the approximate length

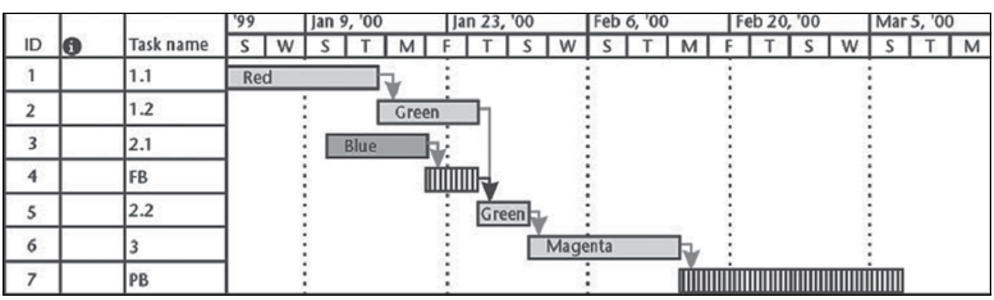


Figure 7.7 The first choice results in a scheduled completion date of March 7, 2000. (Tasks 1.1, 1.2, 2.2, and 3 comprise the critical chain.)

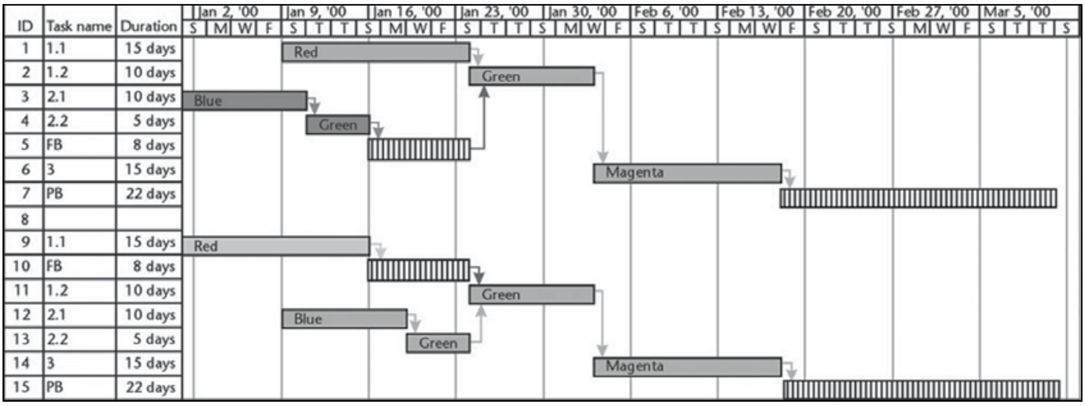


Figure 7.8 The second resource resolution choice leads to the same lead time for the two choices of critical chain and completion on March 10, 2000. (The critical chain for the upper case is tasks 2.1, 1.1, 1.2, and 3. For the lower choice, it is 2.1, 2.2, 1.2, and 3.)

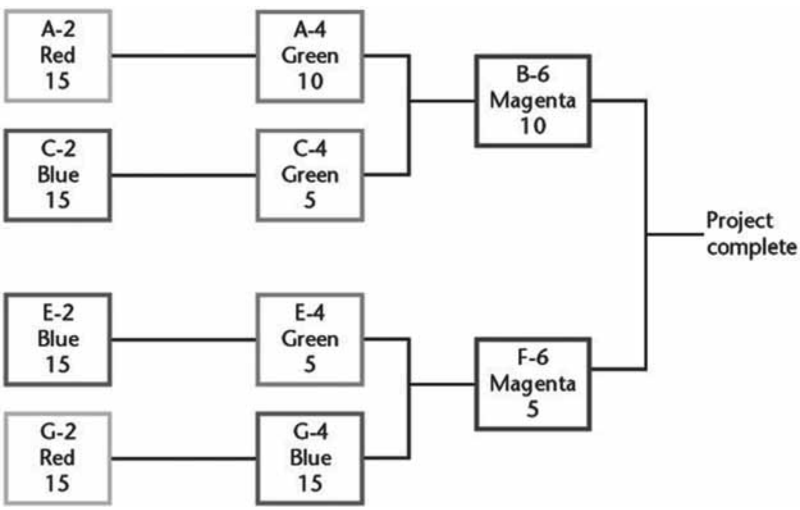


Figure 7.9 Small exercise.

of the schedule that you should have obtained. A good-enough schedule is within a small part of the project buffer.)

7.3.2 Large Example

Figure 7.10 presents the task network for a large example. The top line of each box is an identifier for the task. The color name in each box 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 schedule for this project. The first step is to lay out the network. Figure 7.11 illustrates the project entered in the Microsoft Project Gantt Chart view. Note that Microsoft Project defaults to an early start schedule versus using a late start schedule as I described for the manual method. With the

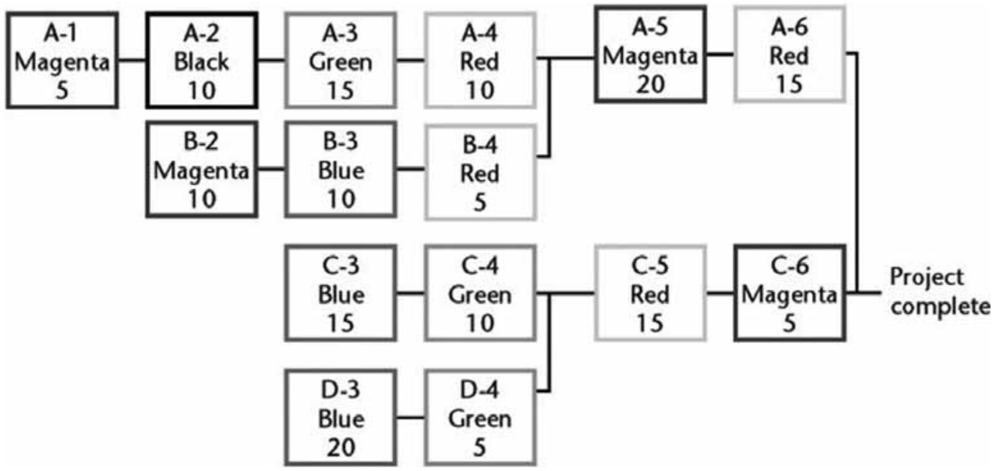


Figure 7.10 Large example. (Task times are already reduced.)

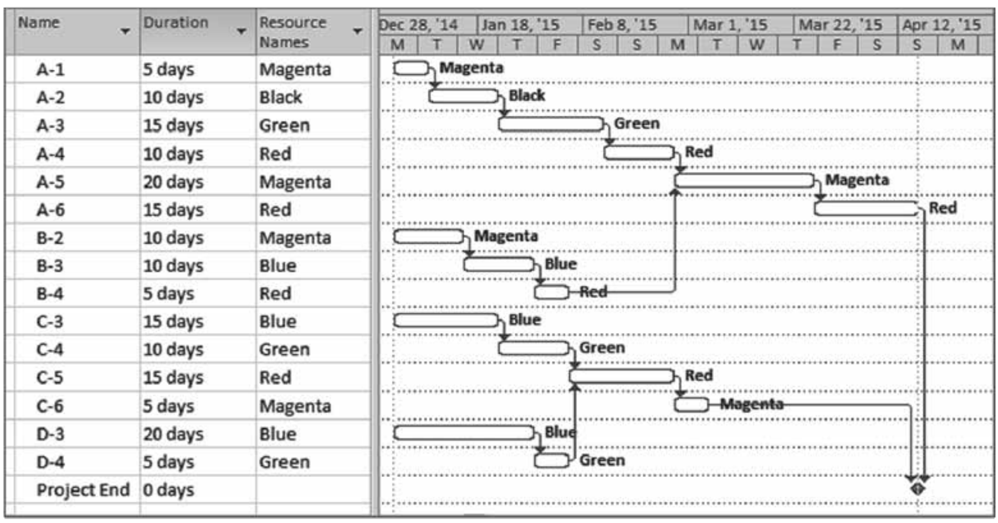


Figure 7.11 Large example Gantt chart illustrates resource overload.

early start schedule, you can look from left to right and notice 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.

Microsoft Project enables you to remove these overloads by resource leveling, as illustrated by Figure 7.12. Note that all chains now have gaps in them, making it impossible to define a critical path.

Next you must identify the critical chain. Figure 7.13 illustrates the critical chain identified by the CCPM+ software. Observe how the chain jumps the logic path several times. This can only happen where there is no task relationship and the tasks use the same resource. 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.

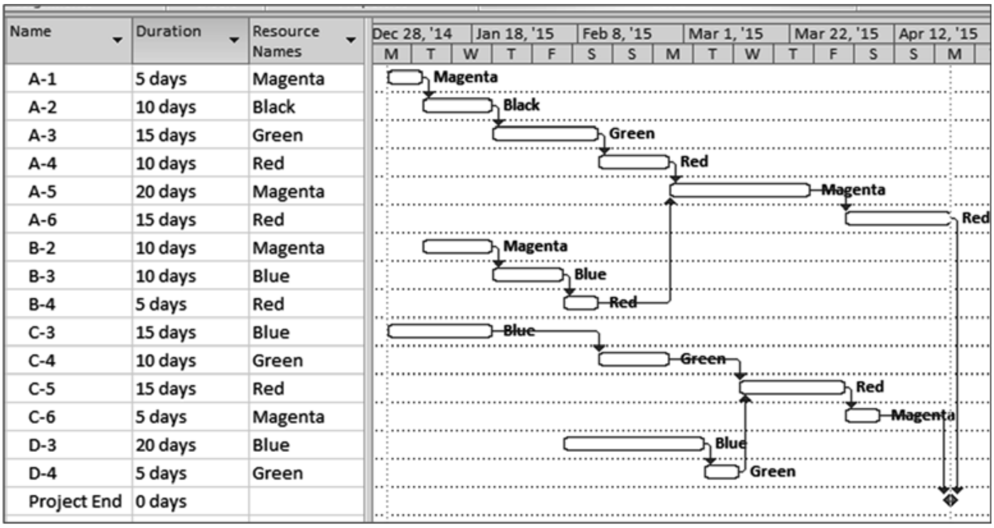


Figure 7.12 Resource overloads leveled.

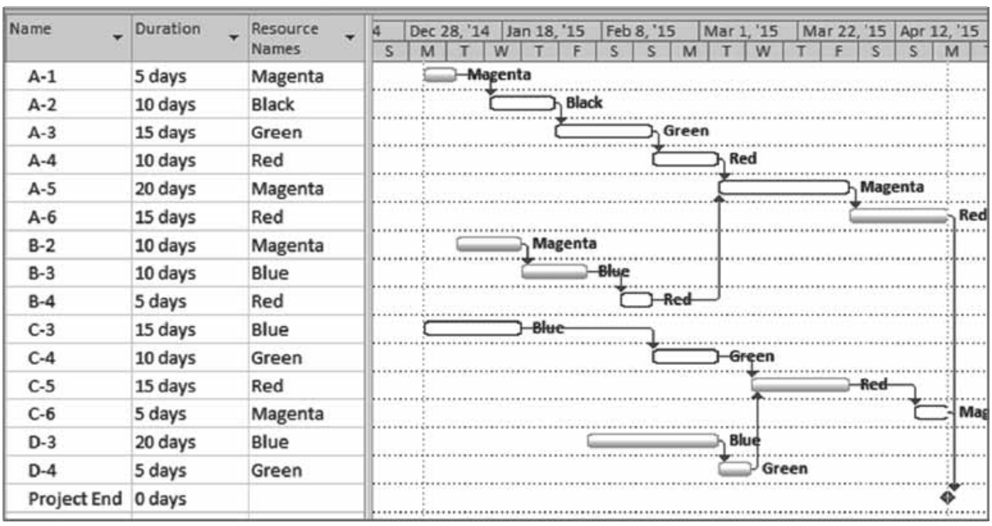


Figure 7.13 Identifying a critical chain (filled bars).

Finally, inserting the project and feeding buffers completes the schedule. Figure 7.14 illustrates the completed schedule. In this case, I sized the project and feeding buffers as 50% of the preceding activity chain. This project requires three feeding buffers.

The following set of figures follows the same process as just illustrated. It illustrates actual task names, a WBS, and more meaningful resource titles. You can consider this project as developing anything that has multiple parts: hardware or software. I added it here in the third edition because I hope it may communicate better to some people and simply to show another large example. Note that Figure 7.15 illustrates dates for each task and no resources. This is usually the case with CPM schedules.

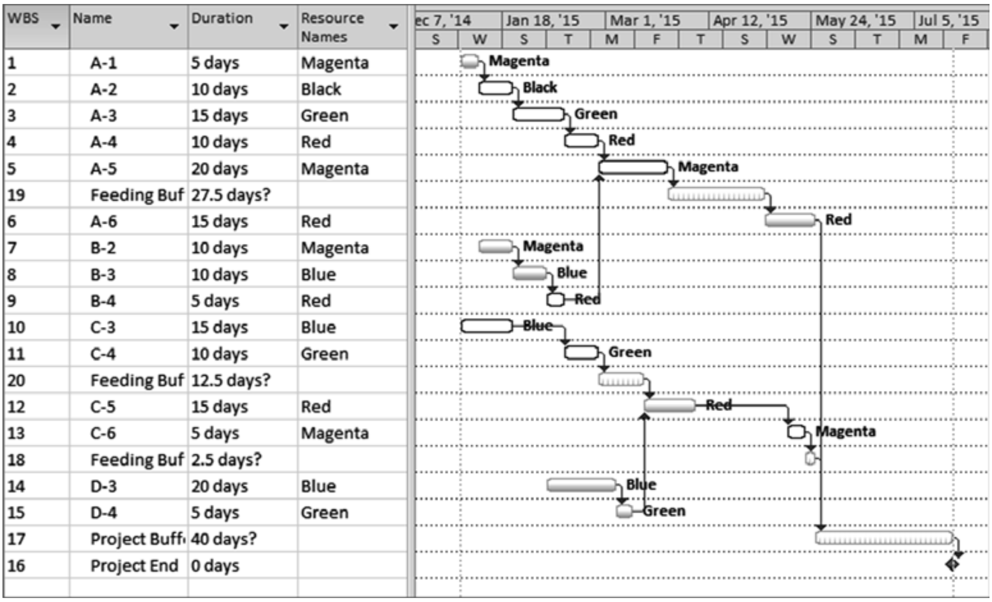


Figure 7.14 Inserting the project and feeding buffers completes the schedule.

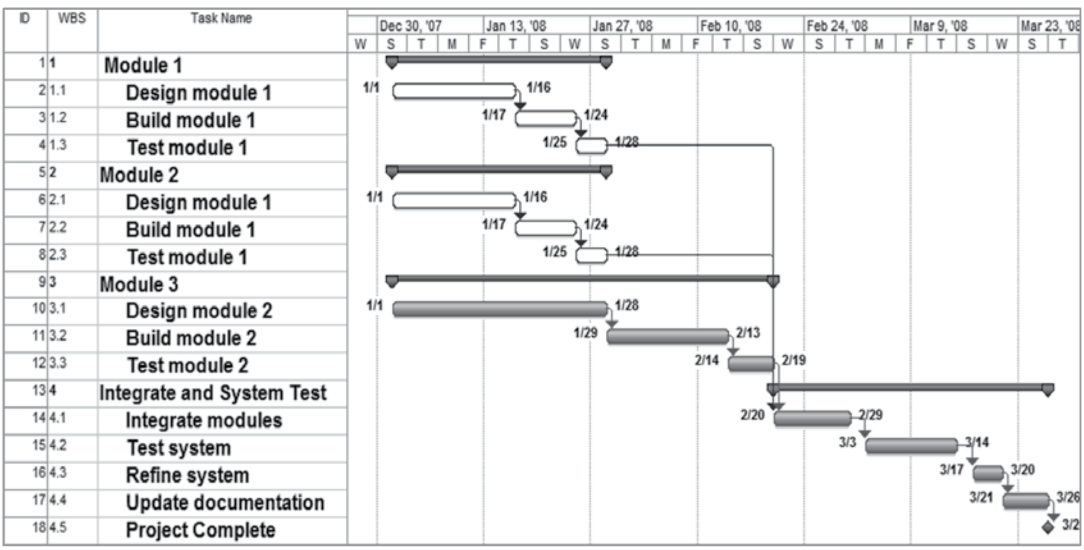


Figure 7.15 An example critical path project schedule, showing the critical path tasks as solid bars.

Figure 7.16 replaces the dates for each task with the resource demanded by the task. Figure 7.16 is also resource leveled. Figure 7.17 illustrates the corresponding Critical Chain schedule with buffers.

Note the gap shown by Figure 7.18 between tasks 3.2 and 1.2. It is normal to have such relatively small gaps in the critical chain after inserting the feeding buffers. In this case, because the task continues on by the same resource, there would be no gap in execution.

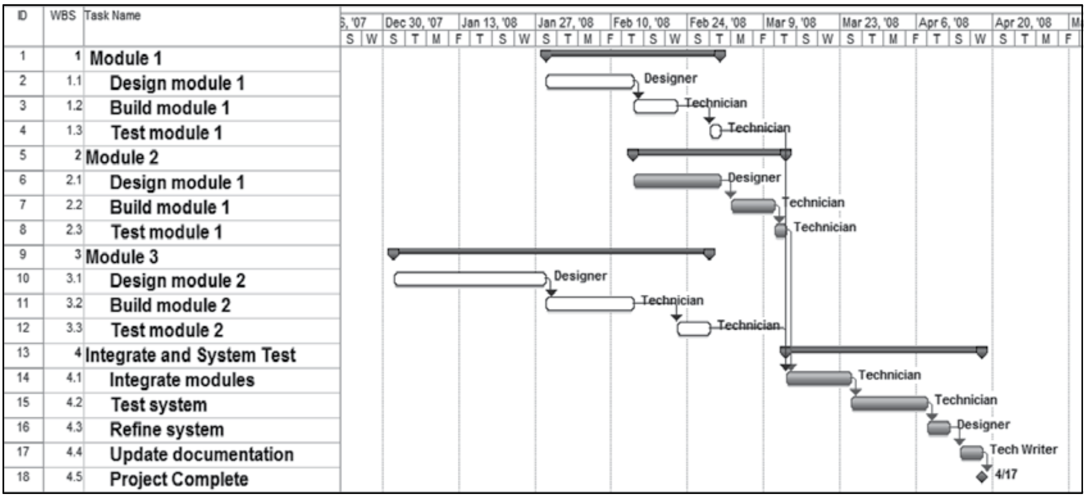


Figure 7.16 Resource-leveled critical path.

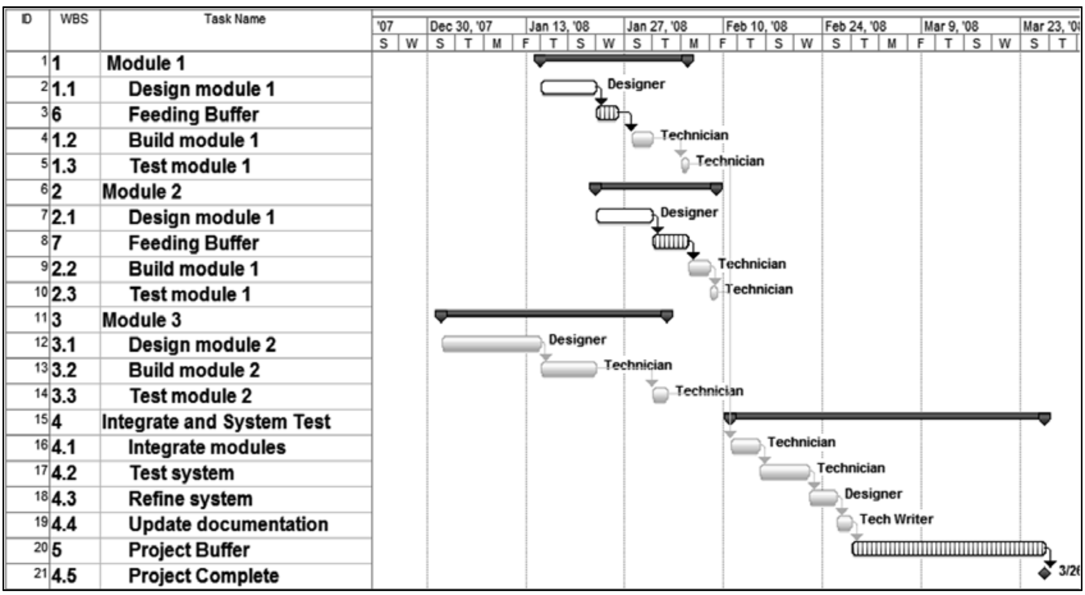


Figure 7.17 The Critical Chain schedule with the solid bars showing the critical chain tasks, the open bars the tasks on feeding chains, and buffers (with vertical lines in the task bar).

7.3.3 Large Exercise

Lay out the exercise in Figure 7.19 as a Critical Chain schedule 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 7.10 for the approximate length of the schedule that you should have obtained.) Note that a good-enough schedule is within the duration of an optimum schedule within a small fraction (e.g., less than 10–20%) of the project buffer.

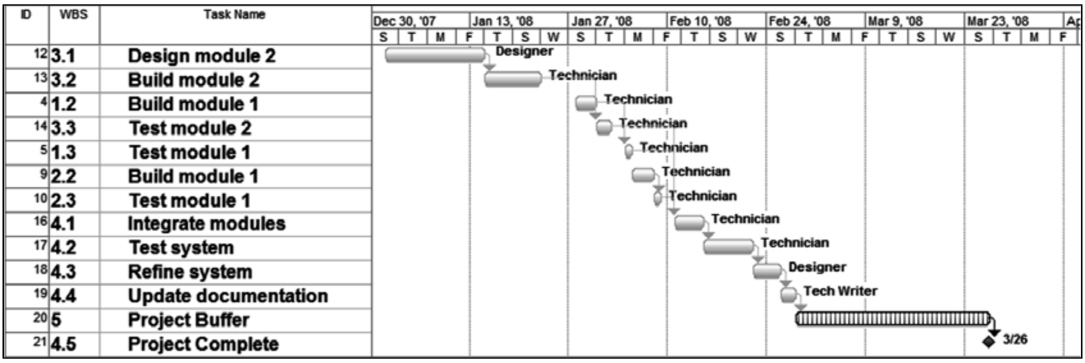


Figure 7.18 Critical chain only for example project. Note that it "jumps the logic" path where there is no arrow connecting tasks (e.g., task WBS 3.2 to 1.2).

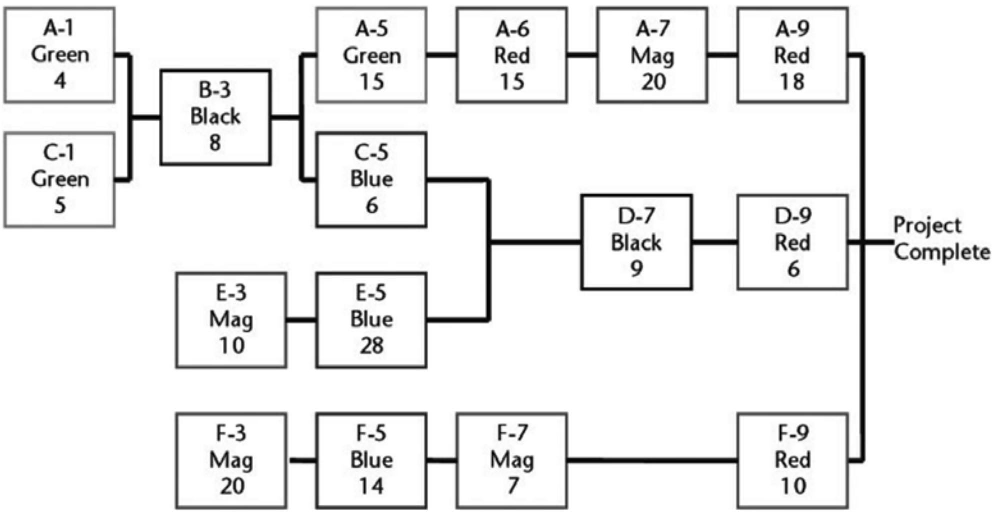


Figure 7.19 Large exercise.

7.4 Buffer and Threshold Sizing

Buffer sizing determines the overall duration of your project and the degree of overall contingency included in the schedule. The buffer thresholds for action determine the frequency with which you will act. I usually set buffer thresholds using a straight line on the buffer consumption chart so that the thresholds are lower early in the project. This provides early warning and prevents allowing the tasks near the beginning of the schedule to use up too much buffer before taking action.

7.4.1 Statistical Background

Recommendations on buffer sizing used statistical thinking to develop relatively simple rules with a supporting theoretical basis. Dr. Eliyahu Goldratt recommended sizing the project and feeding buffers to one-half of the path (chain) length; that is, do not count the gaps in the chain when sizing buffers. The buffers are there to protect the project from variation in performing the tasks on the chain. Some Critical

Chain scheduling tools enable sizing the buffers different ways. I believe the most popular method is the sum of the differences described later. My experience is that it works well.

I believe that Goldratt's method considered 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 task duration estimate times in half. Mathematical justification of his recommendation requires several additional assumptions, some of which I highlight later. 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 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. 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 I give later. 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 variation 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 variation 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 that 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 *D*s. This result is always less than adding the *D*s.

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 1.

The critical path is therefore eight weeks. The critical chain tasks add up to 4 weeks. Because *D* equals 1, *D* squared also equals 1. The sum of *D* squared is then 4, and the square root of 4 is 2. Adding the 2-week buffer to the 4-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 does not happen very often.

7.4.2 Project and Feeding Buffer Size

CCPM practitioners have used a variety of methods to size the project and feeding buffers. I have come to recommend a simple method (half the difference) described later and a method to improve it later if indicated (see Chapter 10). Another method, the square root of the sum of the squares (SSQ), tends to appeal to organizations of engineers and scientists and thus caught my interest for some time. Although the SSQ method is mathematically more complex, experience showed no improvement and a big problem with it. However, as it gained much popularity and may still be endorsed by some, I describe it also.

The following describes several methods that have been used to size buffers. I recommend Method 1.

- *Method 1: 50% of the Differences.* This method sums up the differences between low-risk and mean task duration estimates along the path and sizes the buffer as half of the total. The estimates are in working days and thus do not include gaps in the chain for nonworking time or resulting from resource leveling. For feeding chains that branch upstream, the method conservatively considers only the longest path.

  This method has several strengths: it is simple, it usually provides a large enough buffer, and it allows for long-duration, small-variation tasks such as material lead times. It corrects a weakness of the simpler initial method proposed by Goldratt of using half the chain. That method sometimes caused buffers to be too large when there were long-duration, small-variation tasks in the chain.
- *Method 2: Square Root of the Sum of the Squares (SSQ).* The SSQ method also 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, it uses only the longest chain or the largest result considering each chain.

  The advantages of this method over the initial Goldratt proposal of half the chain is that it allows using known task variation and takes into account how random variation combines mathematically. The primary disadvantage is that it may lead to undersized buffers for long chains.

There are three primary reasons for the underestimate. The first is underestimation of the actual task variation. The second is that the SSQ method assumes that all project variation is stochastic and depends only on the variation in the identified individual tasks. Reality demonstrates that some variation does not satisfy those assumptions. 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. The third reason is that the method makes no provision for the time between tasks waiting for resource availablity. This shortcoming applies to all of the first three methods described here.

- *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 7.1 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.
- *Method 4: Monte Carlo Analysis.* One could use Monte Carlo analysis of the schedule network to size buffers. It can include the advantages of all of the above, other than simplicity, and with the more sophisticated Monte Carlo tools avoid the weaknesses. One would run the analyses and size the buffers to achieve a given probability of project success (e.g., 95%). While this method is straightforward for sizing the project buffer without feeding buffers, it would take some imagination to make it account for and size feeding buffers. The primary disadvantage is the added complexity of the analysis. Because actual task performance probability distributions usually do not exist, there is no reason to expect that the enhanced mathematical processing of the schedule would lead to improved task performance.
- *Additional Guidelines:* The following guidelines help assure an effective buffer:

  1. Seek to have at least 10 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.

Table 7.1 Guidelines to Size the Fixed Portion of Buffers

Cause of Bias	*Range of Time Buffer*	*Range of Cost Buffer*
Resource of successor task busy	50%	0%
Omissions	Some, not to exceed cost impact	5%–10%
Path merging (more than five 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

2. Do not allow any one activity to be more than 20% of the critical chain. *Reason:* The variation 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 50% of the critical chain. *Reason:* Chains with many tasks of uniform length may calculate a relatively small buffer providing inadequate protection.

7.4.3 Buffer Trigger Points

Whenever your buffer penetration exceeds the red trigger, you must 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 two buffer trigger points to plan for management control action (yellow) and to initiate the action (red). 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 must 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 length of the critical chain, commonly called a fever chart (see Figure 7.20). Although you cannot see it here, the upper region is normally colored red, the middle region yellow, and the lower region green. 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 potential variation in the remaining tasks. Another approach is to perform a dynamic recalculation of the buffer required as

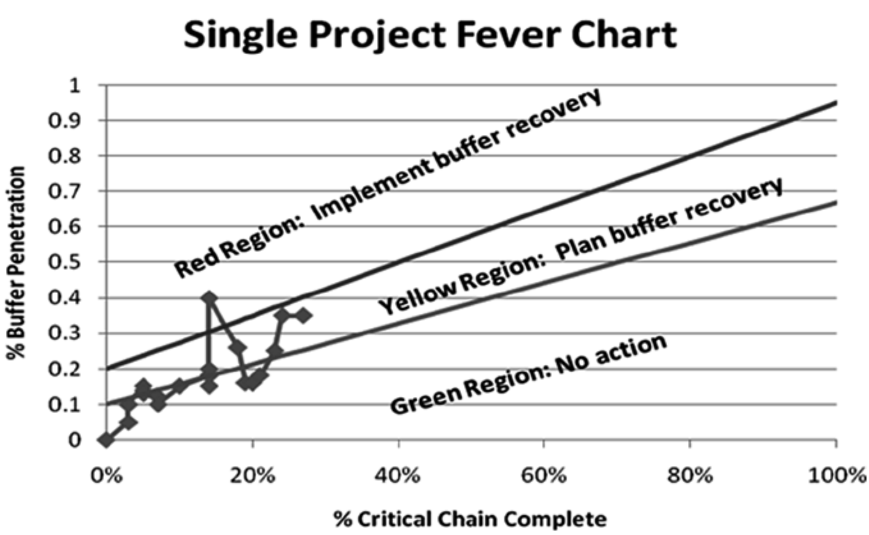


Figure 7.20 Project buffer tracking with a fever chart.

the project proceeds. One Critical Chain software offered this feature. It did not seem to help project performance. For example, the CCPM+ software replaces this dynamic calculation with a linear approximation and in some cases allows the user to adjust the thresholds. Table 7.2 lists typical thresholds.

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.

Many who tried CCPM early originally thought that feeding buffers should be useful as control tools. Experience demonstrates that they can be used more like the shock absorbers in your car: installed and then forgotten. You should set work priority by the effect that tasks have on the project buffer. You calculate this effect for tasks on feeding chains by subtracting the amount of remaining feeding buffer. The CCPM software of which I am aware automates this for you. Although most of the software still provides status information on feeding buffers, you can mostly ignore them in execution. The exception is if all of your feeding buffers are being absorbed early in a project. This indicates that project risk is increasing and may cause you to consider taking action.

7.5 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) should not require a cost buffer. Alas, few organizations are thus enlightened.

Unlike schedule, which sums along paths or chains and not all of the tasks in the project, cost does sum for all of the tasks in the project. Thus, there need be only one cost buffer.

Sizing the cost buffer requires considering a number of factors. First, you may need to 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 or using an overall average burn rate for the project. 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).

Table 7.2 Buffer Threshold Settings

	Buffer Penetration (%)	
Critical Chain Complete (%)	*Green-to-Yellow Transition (%)*	*Yellow-to-Red Transition (%)*
0	10	20
100	67	95

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 scheduling. This method phases the schedule, 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 schedule periodically as it is better defined.

If your organization uses cost-and-schedule-control reporting or uses project schedules to sum up organizational resource demands, you can add the cost buffer into the Project Plan. You can also add it in most schedule software by putting the cost buffer into the project buffer task as a material cost. If you use other means for global resource planning, perhaps you can put it into the buffer as a leveled fixed cost. If you use the individual project schedules to forecast resource demand, then 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 schedule.

Although beyond the scope of this text, I will just mention that aggregating of cost buffers for multiple projects in a portfolio follows the same logic presented above for aggregating the cost contingency for tasks. That is one can have a smaller cost buffer for a portfolio than the sum of the cost buffers required for the individual projects that comprise the portfolio. It is a consideration to keep in mind when approving an annual portfolio to a budget ceiling. It will allow you to approve more projects for the year. Of course, you will control WIP to execute those projects to the capability of your project delivery system.

7.6 Methods to Create the Schedule

People have successfully used a variety of methods to make and Control Critical Chain schedules. The pioneer 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 schedule (i.e., a Critical Chain schedule should have no more than a few hundred activities; less than 100 is preferred).

However, for most organizations, it pays off to use Critical Chain software. The cost of the effort to use manual methods or Critical Path software far exceeds the cost of the Critical Chain software. More importantly, the effort to provide effective task priority in a multiproject environment grows exponentially with the number of projects and project tasks.

It is important to guard against the thought that purchasing Critical Chain software will cause your organization to achieve project execution success. Changes in management and resource behavior are the causes of CCPM success, not software. I have worked with organizations that spent over $1 million on Critical Chain software but were not experiencing success with CCPM. I found they had no understanding of the principles and were making fundamental mistakes such as loading primary resources to tasks at 25% or less and then causing their resources to multitask on four or more project tasks at a time. Once we trained their management to lead improvement, provided the necessary visual controls for reinforcement, and enabled their resources to focus, they dramatically improved their project delivery success.

7.6.1 Manual

The simplest and most commonly used method to manually create work package networks 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 Post-it® 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 overloads.
4. Identify the critical chain.
5. Add Post-it 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 overload and revise calculations.

This process is not difficult for projects with fewer than about 20 tasks. It gets harder after that because 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-overload 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.

7.6.2 Critical Path Software

You may use critical path software to schedule and manage Critical Chain projects. I have had some clients with expert schedulers do this successfully with leading critical path software that does not provide necessary Critical Chain features. The primary missing features are a way to identify the critical chain and buffers. Using such software for effective CCPM requires innovative use of the scheduling software. People that are not full-time schedulers will likely find this quite confusing and tedious. Thus, although it can be done, if your organization lacks enthusiastic expert schedulers, it will be worth your while to purchase Critical Chain software.

That said, I also had the pleasure of knowing one Project Manager in charge of quite large project ($127 million) who attended one 4-hour presentation on Critical Chain and implemented a set of work rules to enforce the necessary Critical Chain behaviors such as relay racer task performance, eliminating multitasking, and focusing supervisors (Task Managers) on having their resource complete a task before moving on to another. He enabled the Task Managers, who had drifted off into a role of accounting for money and worrying that people are busy, to instead focus on solving any problems that were preventing completing started tasks. He used the existing critical path schedule to guide the flow of work and did not explicitly use buffers. He behaved as the buffer manager. His team completed the project early and $25 million under budget.

Most schedule software has 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 network 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 should understand and experiment with these. The Critical Chain method does not depend on the algorithm you use. It simply requires that the final schedule remove all resource overload within the single project. Usually, you can do resource leveling manually if you wish and view the final resource allocations by task. You should not be disturbed by seeming illogical results from resource leveling. The software does what you tell it to do. As long as you use the schedule during execution to prioritize tasks and eliminate multitasking, you will get the desired end results.

After initially leveling resources, you must identify the critical chain. You do this working backwards through the schedule following a logic chain and then switching to a task on another chain whenever a gap presents itself. The task on the other chain must use the same resource. I suggest that you add links to the schedule along the critical chain you thus identify to ensure the resource-leveled critical chain stays 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 or leveling delays 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 10 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.

Finally you need to set up the schedule for execution. One easy way to do this is save a baseline and use a Gantt chart display of the statused working schedule against the baseline schedule. You will be able to directly read off the maximum project buffer penetration by comparing the start of the project buffer task on the working schedule to the baseline. For this to work correctly, you need to allow tasks on feeding chains to have their overruns absorbed by the feeding buffers. One way to do this is to set an actual duration for the feeding buffers to zero or some small number before you start schedule tracking.

While the above provides the information you need for the fever chart, it does not provide the project buffer impact for working tasks and tasks coming up to be started. You need that information for task priority. Successful users of critical path software for Critical Chain scripted a way to use the critical path software float calculation to provide buffer impact data for each task working and yet to be worked. If that sounds beyond the capability of your schedulers, then you should purchase Critical Chain software.

7.6.3 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. Providers have come and gone in this field, so it is not appropriate to list them here. You can find them through searching on the Internet.

Most organizations do not require the most expensive software options that are offered only with large-ticket consulting packages. You will need to train your managers and work with them to get their behavior aligned with the principles of CCPM. If your approach includes the coaching kata (see Chapter 2), your managers can then train the rest of the people in your organization and create the desired organizational behavior through ongoing coaching feedback.

7.7 External Constraints

Projects may have external constraints. These factors may influence the project lead time and may not be 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. However, you should consider 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.

7.8 Reducing Scheduled Time (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 scheduled 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. Instead, rework your project network as described below to obtain a schedule that meets the required end date with a suitable project buffer.

7.8.1 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 overloads 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 schedule 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 schedule, 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 schedule.

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 schedule 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 schedules 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.

7.8.2 Acceleration with Increased 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 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 with which I recently worked, one day's acceleration could mean $10 million and $18 million, respectively. Any payment made to accelerate was immediately worthwhile.

7.9 Preparing for Project Kanban

If you use Project Kanban to control the flow of work at the work group (Task Manager) level, you need a way to link the CCPM schedules to the Kanban boards. You need to provide the task card and priority information to the boards and feedback the schedule status information from the boards to the CCPM schedule. Most scheduling software allows you to filter and sort the tasks to facilitate the export process and then export the task information in various file formats. Some Kanban software allows you to import the task information files exported from the schedule software.

7.10 Frequently Asked Scheduling 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. *Key people in our organization will never be able to stop multitasking. They are the experts in technical areas and thus the go-to people whenever management needs something urgently. How do we handle that?* Many people in many organizations have made that claim and found ways to resolve it. For larger organizations, one way to handle it is to not put these people in as resources in the schedules but rather deploy them to help more junior resources when project tasks move into the red (action) region or when urgent nonproject work must take first priority. Others have set aside resources to handle the unplanned work. Smaller organizations may need larger buffers in their project schedules to account for work stoppages when the resources are not available. However, it is always beneficial to the people and the organization to work to eliminate multitasking. When a critical resource simply must be pulled from working on a project task, they should focus 100% on that alternative work until it is done and they can come back and focus 100% on the project tasks.

2. *After we add the feeding buffers, is it 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.
3. *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.* Your schedule development process should examine your schedule logic and resource loading to try to eliminate such gaps. Consider where the noncritical chain feeds the critical chain and the relative variability of the two chains. Remember that we are subordinating everything else to the critical chain. Gaps in the schedule 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, pay extra attention to tracking progress on these parallel paths because you will not have the feeding buffer measurement to help.
4. *Why do we not 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.
5. *Our schedules have thousands of tasks. How can we schedule 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, task checklists, or subprojects where more detail is required. As noted earlier, you should also limit total single project duration, including the project buffer, to no more than 1-year duration. You do that by breaking the large project into overlapping smaller projects and treating the group of projects as a multiproject program. Using that approach, CCPM projects have been successfully completed with more than 30,000 tasks in the entire multiyear project.

   Be sure to use 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 turnaround times for schedule updates and the loss of control.
6. *We have tasks in our project schedule 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 2, 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. You might use a buffered milestone for your input to them to ensure that you deliver their inputs on your agreed-to dates. Other unique situations demand unique solutions. You should use the five-step focusing process in those instances.

7. *Our (management/client) has specific intermediate milestones for which they want us to schedule a date 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, I 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, I suggest you first try to convince your management or client that buffer reports are actually a better tool. Failing that, I suggest that 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 I again suggest option one.)

8. *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.

9. *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 schedule (i.e., not your worst case). For example, include tasks to perform the work that you most often find from such investigations. You should also structure your schedule to reduce risk by determining the additional work as soon as possible in your schedule. This may require linking tasks in such a way as to not late-start chains of tasks that include the discovery work. For example, tie all the discovery work into a milestone that is the predecessor to all follow-on work in the project.

   Some software may allow you to add tasks to your project as you go, but generally this is not possible because it might redefine the critical chain and thus feeding buffer locations. You have to decide when it is time to reschedule. Once your project buffer penetration exceeds 100%, and you cannot come up with a realistic buffer recovery plan, it is time to reschedule your project and resize the buffers. Rolling wave planning may also be a solution to this problem.

10. *Our Critical Chain software places 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 that you gave it and builds a resource-feasible schedule. If you have not told the schedule 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).

11. *How does this chapter relate to the three rules?* The schedule prepared using the methods of this chapter supports Rules 1 and 2: Focus and Buffer. Focus comes from performing one task at a time. Identifying the primary task resource at 100% and resource leveling the schedule provides a baseline schedule to enable focus. Of course, actual task durations will not match the estimates, so in execution we will use the prioritized sequence of task performance in place of the initial sequence, but the placement of buffers ensures that all of the tasks can be performed in a focused manner within the buffered schedule duration.
12. *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 in Table 7.3, they are good enough.

7.11 Summary

This chapter has described how to create a Critical Chain schedule for a single project. The steps of creating a network 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 scheduling 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.

Table 7.3 Exercise Results

Project	*Critical Chain Length*	*Project Buffer*	*Total Project Lead Time*
Small exercise	50	25	82
Large exercise	107	47	154

2. You must resource-load and resource-level CCPM project schedules 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 the 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 duration estimate differences (low risk duration minus mean duration) for the tasks along the chain.
4. Project buffer trigger points determine the need for control action. Decide on the trigger points as you prepare your schedule for execution.
5. The (optional) cost buffer provides aggregated cost protection in an analogous 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 schedule is a relatively small addition to the work necessary to construct an effective Critical Path schedule. It may be less work and create a more useful schedule if you reduce the level of detail in your schedule. Even when it requires work that you have not done before (e.g., resource-loading your schedule), the benefit far exceeds the investment.

Reference

Advanced Projects. (2013). CCPM+ Software. www.advanced-projects.com. Last accessed November 2013.

CHAPTER 8

Developing the Multiproject Critical Chain Plan

8.1 The Multiproject Constraint

This chapter addresses the new rule 3: Pipelining. The Theory of Constraints (TOC) provides one point of view on the multiproject environment. From that perspective, we first ask: What is it that constrains the enterprise from completing more projects or completing the existing projects more quickly? 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 (also known as triple constraint: scope, schedule, cost) and do it in a way that allows focus on increasing the project throughput of the enterprise? How do you exploit the constraint?

Consider a reference environment with which most people are familiar: mowing a lawn. Think of 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. This requires the operator to stop mowing and clean out the mower and restart it before starting to mow again.

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, one way to exploit the system constraint is to use feedback from the system to adjust the rate of processing. You might listen and slow down the lawn mower as the engine begins to slow down. You complete the mowing much more quickly by exploiting the capability of the system. Alternatively, you might exploit the constraint by raising the cutting height to match the processing rate to the feed of the work or adjust your mowing schedule so the lawn does not grow as much between mowings. You might need to subordinate your mowing schedule to the system capability to do this. In other words, you adjust the demand for work from the lawn mower to something that the lawn mower system can handle. Normally, you do not run right out and purchase a new lawn mower or rebuild the one you have, although later on you might do so to increase throughput (e.g., also mow your neighbor's lawn).

The organization counterpart to buying a new lawn mower is hiring more staff, and efforts such as reengineering are similar to rebuilding the lawn mower that you have. A simple adjustment (raising the cut height of your lawn mower) gets you back in business with greater throughput. Likewise, a simple adjustment to how your organization plans and releases projects can have a huge impact on the bottom-line result.

You might also consider this from the Lean perspective of reducing waste. Restarting your stalled lawn mower was waste. Reducing the demand on the lawn mower to what it could handle eliminated that waste.

Ohno (1978) recognized the need for production leveling as key to the path for waste reduction as early as the time of the Korean War (p. 12), noting: "We looked for ways to level all production." His Kanban system provided a tactical tool for daily use within the production workflows. Demand leveling at the factory level became known in the Lean community as "heijunka" (Pascal, 2007), providing a way to reduce the waste resulting from mura: the unevenness in of fluctuations in the flow of work. While all of the thinking and effort put into production leveling in the Lean community for production applies with little modification to project work, including knowledge work projects, the concepts have yet to fully bridge the gap between production and project management. Perhaps you and I can help complete that bridge.

PMI acknowledges the need to consider capacity constraints when planning project portfolios (PMI, 2013). The works that I have reviewed sometimes use the word pipeline, but it is from the perspective of ensuring enough projects staggered over time to sustain the new product flow, not from the perspective of demand leveling. I have found little in the Project Management Body of Knowledge that addresses demand leveling as we address with Pipelining, other than articles I wrote (e.g., Leach, 2005 pp. 372–375, and, of course, earlier editions of this work).

Figure 8.1 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, the three projects are on schedule to be completed on 5/21.

For the projects in Figure 8.1, assuming that these projects are all the same, the resources have to be divided among the three projects even if you have only one resource of each type. Thus, either the project schedules assume multiple resources of each type are available or that the resources will multitask across projects (i.e., are working only a fraction of the time), and 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 is, the whole organization cannot complete projects any faster than the most loaded resource can complete its project tasks. That most loaded resource is the capacity constraint of the system, which we designate as the drum resource.

Some companies check resource availability across all projects. They then argue to increase resources (buy more lawn mowers). This is moving to the elevate

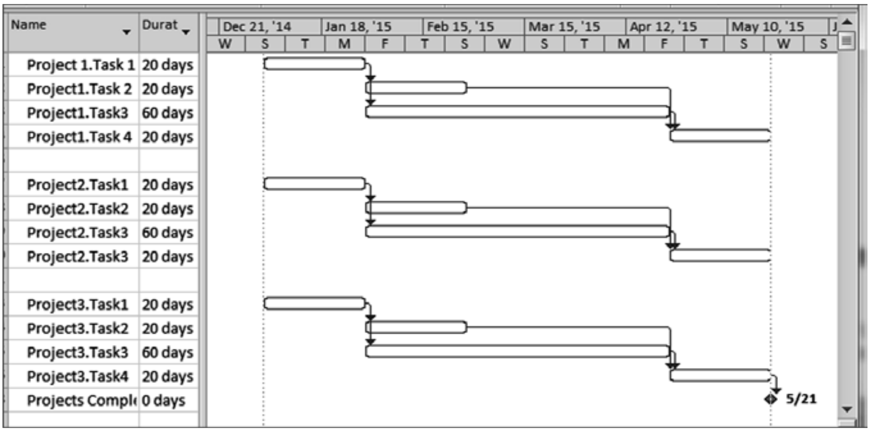


Figure 8.1 Three projects in a multiproject environment.

stage of the TOC before completing the identify, exploit, and subordinate steps. It is a very expensive strategy.

To improve throughput, you have to first identify the company capacity constraint. This was originally thought of as a resource: a certain type of skill with limited availability. It was quickly realized that it may be a physical or policy constraint. Using terminology that comes from Dr. Eliyahu Goldratt's production methodology where the drum sets the beat for the entire factory, the company constraint resource was designated the drum for scheduling multiple projects. Here the drum sets the beat for all of the company's projects. Think of the drummer on a galley or racing shell. What happens if even one rower gets out of beat?

The project system becomes a pull system because the drum schedule, which we call the "Pipeline," determines the sequencing of projects. You might recall pull as a 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.

Figure 8.2 illustrates the CCPM method. Compared to the previous critical path case, CCPM reduces each activity time to one-sixth of the original 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. In this example, the CCPM approach identifies the resource supplying activity 3 as the capacity-constraint resource: the drum. CCPM exploits the resource by synchronizing the projects using this resource as the drum. CCPM subordinates to this resource by adding capacity constraint buffers (CCB) between the projects. The capacity constraint buffer ensures that the capacity-constraint resource is available for the subsequent project. The CCB is not derived from the individual Critical Chain schedules. Section 8.3.5 describes how to size and treat CCBs.

Figure 8.2 shows the CCPM schedule completing the three projects (including the project buffer) April 1 as compared to May 21 for the critical path schedules. It shows the first two projects completing much earlier. Based on what you have learned for single projects, you can expect the CCPM projects to not use all their

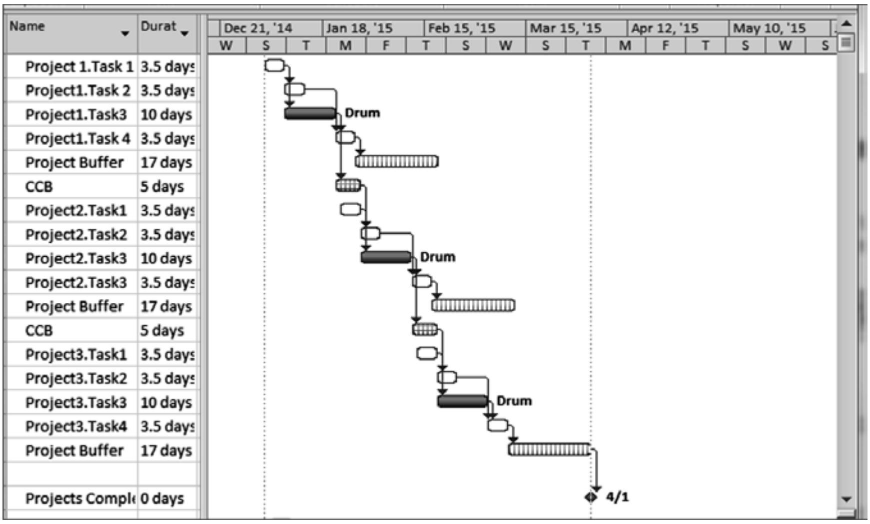


Figure 8.2 CCPM multiproject plan reduces project duration and increases project throughput.

buffer and thus be earlier still. The two projects that have their starts delayed still finish sooner. 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 Managers prioritizing resource assignment to 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. Such attempts never work. I hope that by now you understand the reason why seeking to schedule each task to dates 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. Your schedule is only one possibility 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 protective capacity. Those upstream of the constraint resource must have protective capacity to ensure that the constraint resource is never starved for work because 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 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 demand resources in any order, there tends to be a similarity in the order within a company based on the type of projects that 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 protective capacity and/or the larger buffer that it needs to not influence the overall lead time.

8.2 Improving Throughput at the Multiproject Constraint

The constraint resource becomes the drum for the company projects (like the drummer on the ancient galleys 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 or one that is not easily elevated.
   - 1.2. The constraint resource is not necessarily the same as a specialty skill that is in short supply and is occasionally overloaded. The constraint resource is usually a resource that is heavily used on most projects. Local bottlenecks occur when an organization has only one or two resources with a unique skill. Recurring local bottlenecks are often relieved by mentoring and cross-training rather than by hiring additional resources.
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. This is for Pipelining only and does not affect execution behavior.
   - 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. Insert capacity-constraint buffers (CCB) between the individual project's schedules, ahead of the scheduled use of the constraint resource. Delay the scheduled start of each lower-priority project to accommodate the capacity constraint buffer. This protects the drum (constraint) schedule by ensuring that we have the input ready for it.

3.3. If inserting the capacity constraint buffers overloads the constraint-resource schedule, resolve the overloads by delaying project starts.

4. Elevate the capacity of the constraint resource.
5. Go back to step 1, and do not let inertia become the constraint.

The following sections describe the features of this process.

8.3 Multiproject Critical Chain Features

8.3.1 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 by dividing the project throughput (usually in local currency) 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.

Keep in mind the TOC good-enough concept. With Pipelining, all projects should finish sooner. So priority or sequence matters mostly because it increases throughput. Projects that are sequenced to start later in the sequence still finish sooner than they would have if the projects had all started at once because multitasking would make all projects take much longer.

8.3.2 Select the Drum Resource

The drum resource must be shared across all the projects you consider part of a single multiproject environment for Pipelining. 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, but only one drum per independent resource group.

Alternative 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 resources appear as temporary capacity constraints. For example, suppose a number of projects happen to demand a particular resource at one time, thereby temporarily exceeding the resource's 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 schedule and control system must handle the fluctuations 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. That is, use the capacity constraint resource only for work that demands their special expertise. 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 constraint and therefore must become the drum for all of the projects.

Because the purpose of selecting the drum resource is to sequence the starts of projects and avoid overloading the system, it usually does not matter much if you select the wrong resource as the drum. As long as you use a highly loaded resource, you will still get some degree of project sequencing. 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. CCPM schedules provide 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 did not recommend this method for production because he claimed 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.

To achieve the maximum effect of staggering the projects the drum resource should be a 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 portfolio.

Many companies choose to identify resources by individual names. Unless those resources are truly unique, it is not a good practice. Some feel that the resources are so highly specialized that they cannot do otherwise. If this is true, then you have no other option. In this case, 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.

8.3.3 Nonhuman and Virtual Resources

Your constraint resource does not have to be a human resource skill. It can be a piece of mechanical equipment (e.g., a crane or test facility). Your organization might have multiple units of that resource yet find its use often in contention. Project scheduling software does not know or care if the resources you enter are human or nonhuman. You can do the Pipelining process the same way.

Some organizations feel that their project flow is not constrained by a particular type of resource but rather by a type of activity that might draw on multiple resources for short periods of time, for example, system test and bug fixing in software development. In this case, the resource can be virtual (e.g., something named system testing and you may allow only so many system tests at a time in your organization). Once again, the software does not know or care what the thing is that you put into the resource field so you can put in something like a system test. Some in the TOC community like to call this a "virtual drum," but in my view it is better called a "virtual resource" or simply "the drum." If you use it to sequence projects, it is a real drum for your organization.

8.3.4 The Drum Schedule (Pipelining the Projects)

The drum schedule sequences projects through allocating the drum resource across all projects. The manager who has responsibility for the drum resource usually manages it and one or more Master Schedulers perform the technical functions of Pipelining. The drum schedule determines the system capability to process projects. It deploys the WIP limit for active projects and provides the start date for each project.

The TOC community calls 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; do not overfill the pipe.

The Master Scheduler needs the drum resource demands for each project and each project's priority to create and update the Pipeline. The individual Critical Chain project schedules determine the duration, earliest time, and relative times for each of the drum-using activities in each of the projects. Figure 8.3 illustrates the drum-resource demand from three projects, A, B, and C, positioned from highest priority on the bottom to the lowest priority at the top. The drum schedule must fit

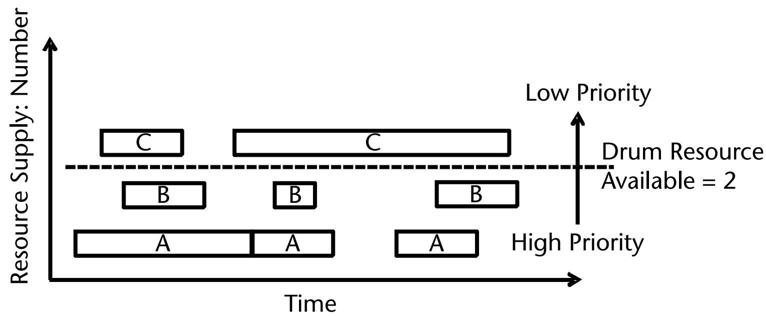


Figure 8.3 Three-project drum-resource demand, assuming that all three projects are to start today.

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.

Note that the drum-resource use cannot be scheduled earlier than shown in Figure 8.3. 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. Because 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 8.4 illustrates the resulting drum schedule. Note that the capacity constraint buffer does not appear in any project schedule. It just helps set the start date for the projects.

8.3.5 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.

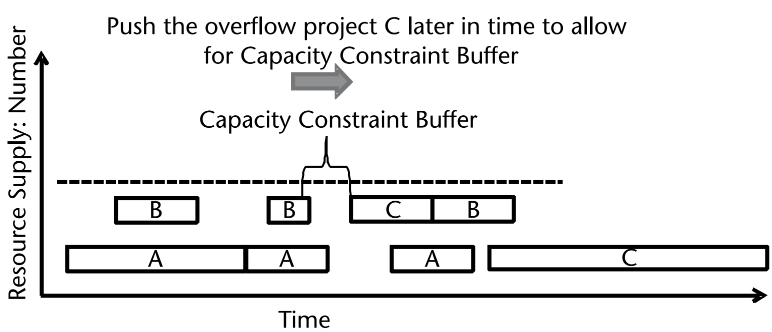


Figure 8.4 The drum schedule accommodates all project demands, including capacity-constraint buffers.

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 supply of the resource. This fits well with CCPM because we know that those apparent overlaps are not real; they are an artifact of presenting a reality with variation using a deterministic illustration, the bars on a Gantt chart.

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. You need to add for that to account for people's nonproductive time (e.g., vacation, sick time, and administrative time). You may need to add for it to account for time spent on nonproject work. If you do not size the capacity constraint buffer large enough, you will have too much WIP and 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 queue 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 that they can dissipate rapidly when extra servers are applied.

I ask project-management groups, "Suppose that the average rate of processing 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 that stores close the doors at night.

Figure 8.5 repeats the classic queuing curve for one line and one server that I illustrated in Chapter 3. 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 that the average arrival rate equals the average processing rate. 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 net

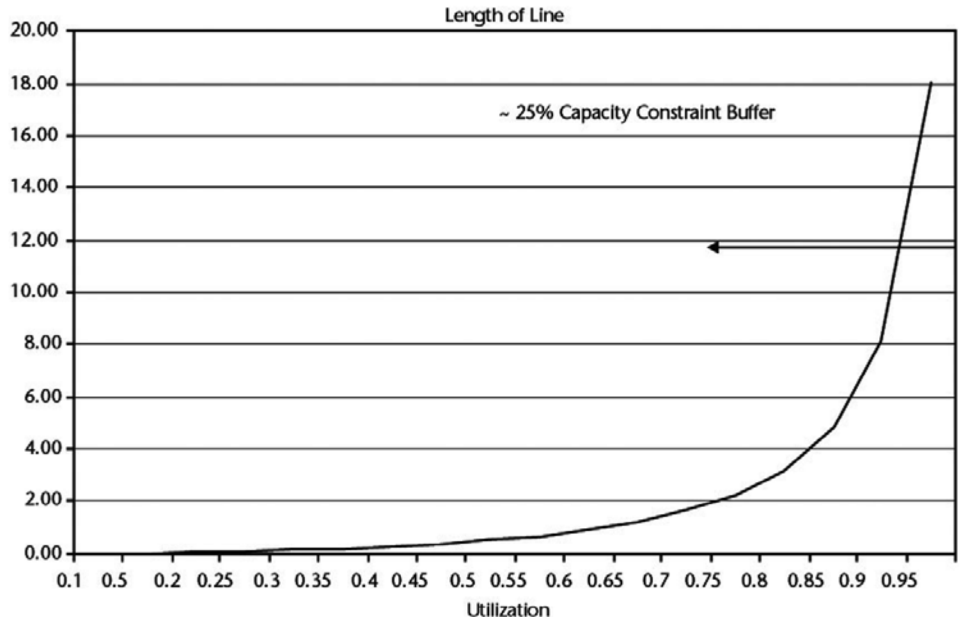


Figure 8.5 The queuing model predicts an infinite line when the average arrival rate approaches the average processing rate. [For project resources, utilization = (average arrival rate of tasks)/(average duration of tasks).]

available resource. To turn this into a capacity-constraint buffer size, you need to add to this 30% for time the resources do not actually spend on project work. Also note that utilization refers to the net available resource so be sure to use the right reference base.

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 queuing curve provides a second equation for Little's law: the relationship between WIP and cycle time. Recall that Little's Law related the Work in Progress (WIP), cycle time (C), and throughput (T) as:

$$T = WIP/C$$

Figure 8.6 illustrates the queuing curve with the axes relabeled to relate to Little's law. Figure 8.6 shows three queuing curves for different levels of variation in the process. The curves move up and to the left as variation increases. C is a measure of variation that increases with variation. The arrow illustrates what happens as you reduce WIP. In addition to reducing the cycle time of projects in a highly nonlinear and beneficial direction, it also likely significantly reduces variation in the processes moving cycle time even lower. Reducing WIP is leveling demand. Pipelining with a capacity buffer provides a way to permanently implement WIP reduction.

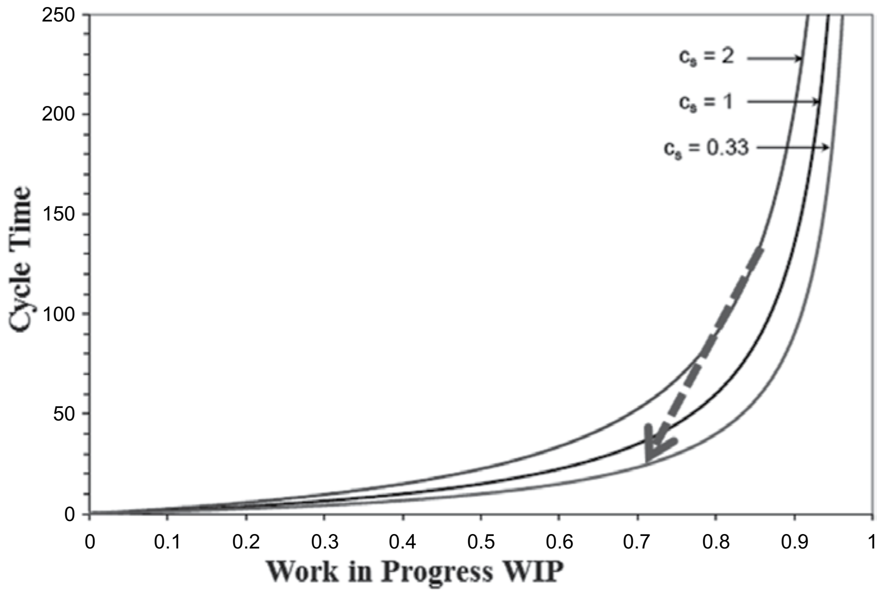


Figure 8.6 Relating the queuing model to Little's law shows how WIP reduction greatly reduces project cycle time.

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 pre-overtime levels.

If you do not want your projects to wait an infinite amount of time for the drum resource, you must use a capacity-constraint buffer in the range of at least 25% to 30% plus an allowance for nonworking time during the year. This suggests a capacity constraint buffer of at least 50%.

It is also desirable to load the system on the flat part of the curve to have a robust scheduling system. When perturbations occur, even special causes in some instances, they have minimal impact on project cycle times if you schedule to keep the drum usage on the flat part of the curve. Even at 75%, upside variation in demand (utilization) can lead to significant schedule impacts.

8.3.6 Nonworking Time

Some organizations apply scheduling policies to schedule resources to a maximum of 70% to 75% on a task. They do this to account for the fact that over the course of a year people spend time on things other than project work such as vacations, sick time, and performing administrative work. I recommended earlier that you always schedule the resource that determines the task duration at 100%. You can add an allowance to the capacity constraint buffer to account for this nonproject work time. When you do it for the drum resource, it means that you are allowing even more flexibility for other resources. You can also include an allowance for the average loading of nonproject work.

It is easy for the nonworking time to approach 25% of the time available in a year. Allowances for nonproject work time can also be substantial. You need to add these allowances to the 25% allowance for queuing. Usually this means a capacity constraint buffer of at least 50% of the time of the drum resource.

8.3.7 The Drum Buffer

Goldratt envisioned a 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 idea was like a feeding buffer. The idea was to place the drum buffer in the project schedule immediately prior to the activity using the drum resource. Do not confuse Goldratt's drum buffer with the capacity-constraint buffer. The capacity-constraint buffer is not visible in individual project schedules; CCPM uses it for project staggering. The drum buffer would go directly into individual project schedules. It could directly affect the project's duration if it is on the critical chain.

Experience shows that the drum buffer is not necessary. There are usually enough projects that any highly loaded resource will not be idle and using the prioritized task list, as discussed in the Chapter 9, eliminates the need for it. I only mention it here in case you hear of it. Some people confuse it with the capacity constraint buffer.

8.3.8 Project Schedules

Once you have the drum schedule, you set the individual project schedules in time sequence by aligning the start of the projects to match up the drum-using activities while maintaining the capacity constraint buffer. You can think of it as working backwards from the drum-using activity to schedule the start of the project. Because you had to have the project Critical Chain schedules with a "time now" start date to create the drum schedule, this amounts to delaying the start of some project by the amount that you had to delay the drum-resource-using activity to fit it into the drum schedule. You move the entire project schedule to do this.

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 sequence the project starts to level the drum resource demand 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.

8.4 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 with which I have been engaged 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., this 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 car completes the race. This a long race with drivers who switch out. The drivers of the car 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 race car comes into the pit, every resource must have the tools and supplies that 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 and then working to reduce the overall duration for each project. Next you pipeline (sequence) the projects to the most constraining resource (drum). Nevertheless, you are not done. The next set of tasks to exploit the constraint requires you to act like members of the pit crew and prepare detailed plans to maximize effectiveness when the car pulls into the pit.

8.5 Introducing New Projects

New projects 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 may 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 projects 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 8.7 illustrates the introduction of a higher priority-project (D) 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 8.7. 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 8.8. 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.

8.6 Example

This section provides a larger and more realistic illustration of one way to perform Pipelining. The projects used are relatively small and we are only Pipelining four projects in the example but the features in the schedules and the method apply to larger projects and Pipelines of about 20 projects or less. More projects require a different approach. Some CCPM software automates the Pipelining process, but fairly large implementations also use simpler methods to limit overall project WIP such as Excel spreadsheets. This example uses Microsoft Project.

The first step is to create a Critical Chain schedule for each project. The projects must draw from a common resource pool. This is necessary to ensure that you can look at the summary demand for resources from all of the projects at once. Even if you use exactly the same name for resources, some scheduling software will consider them as different resources if they come from different resource files. Figure 8.9 illustrates the simple project schedule used in this example. In these illustrations,

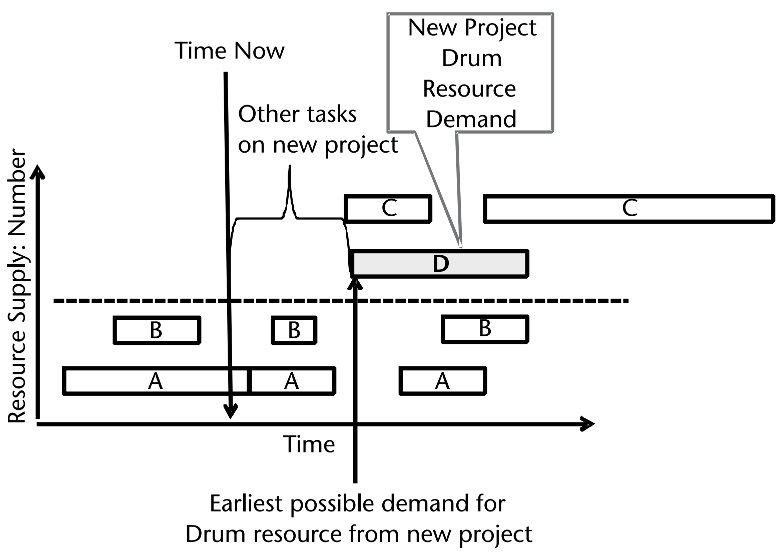


Figure 8.7 A new project is added to the drum demand and judged by management to be higher in priority than an ongoing project.

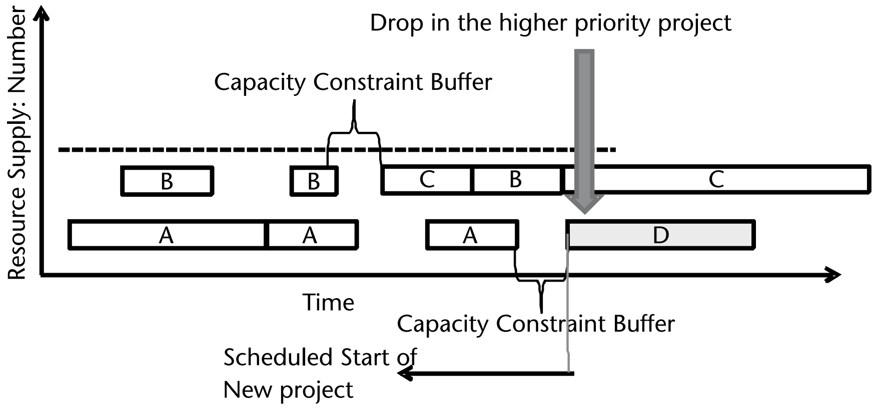
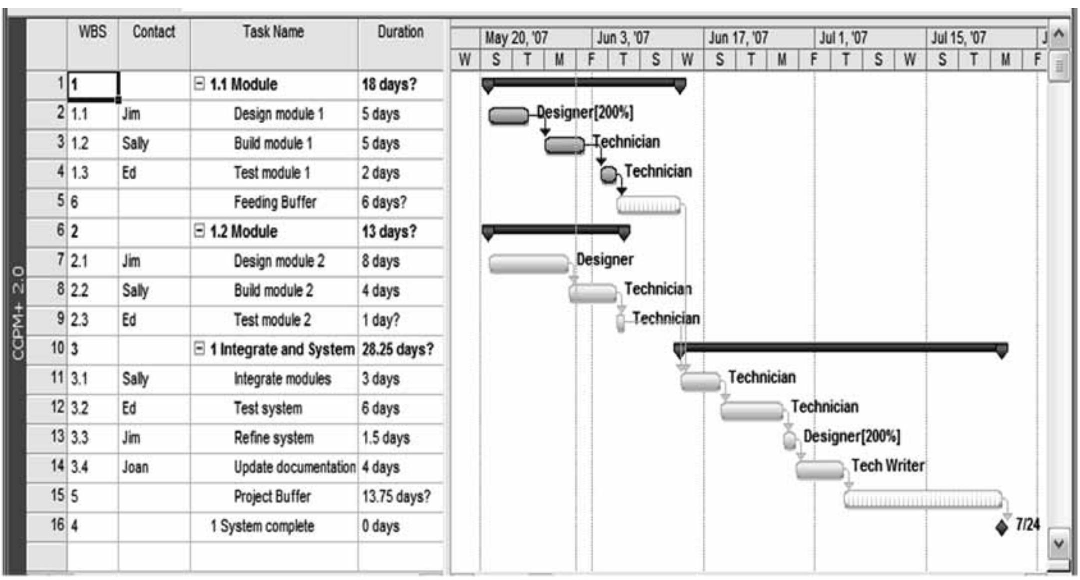


Figure 8.8 Resolving the drum demand sets the schedule for the drum resource in each project, including the new project.



	WBS	Contact	Task Name	Duration
1	1		⊟ **1.1 Module**	**18 days?**
2	1.1	Jim	Design module 1	5 days
3	1.2	Sally	Build module 1	5 days
4	1.3	Ed	Test module 1	2 days
5	6		Feeding Buffer	6 days?
6	2		⊟ **1.2 Module**	**13 days?**
7	2.1	Jim	Design module 2	8 days
8	2.2	Sally	Build module 2	4 days
9	2.3	Ed	Test module 2	1 day?
10	3		⊟ **1 Integrate and System**	**28.25 days?**
11	3.1	Sally	Integrate modules	3 days
12	3.2	Ed	Test system	6 days
13	3.3	Jim	Refine system	1.5 days
14	3.4	Joan	Update documentation	4 days
15	5		Project Buffer	13.75 days?
16	4		1 System complete	0 days

Figure 8.9 The projects used in this Pipelining example are as shown by this figure.

the Task Manager is identified in a field labeled "Contact," a default field available in Microsoft Project. You can change the field name to Task Manager.

You should next ensure that each project schedule is as short as possible. With CCPM software, you can use filters to see just the critical chain (see Figure 8.10). Look for gaps, such as the one after task 2.3. In some cases, changing resources available to the single project or changing the logic may reduce or eliminate the gap. In this case, the network requires the gap for the feeding buffer. The gap does not matter during execution but is necessary to ensure enough time to reduce task synchronization delay.

Check that tasks on the critical chain are worth delaying the project result. If not, adjust the logic to take them off the critical chain. Check if the critical chain

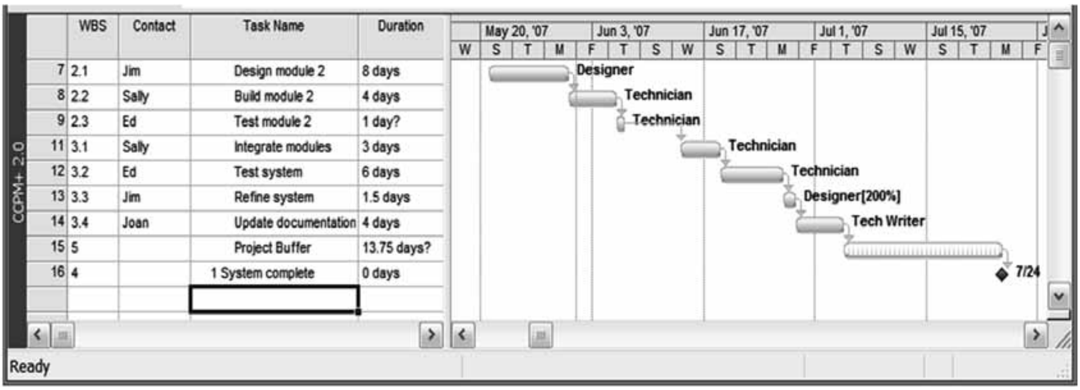


Figure 8.10 A filtered view to show just the critical chain and project buffer, sorted by start date, helps identify gaps in the critical chain.

has any unnecessary resource delays, and consider changing the method of resource leveling if you see a way to shorten the critical chain.

8.6.1 Pipeline

Insert the projects into one file. Figure 8.11 shows four projects for this example. It shows only the summary tasks for each project, but the full project detail is there, just hidden. Note that I have also inserted a column to the extreme left for Project. That will help you identify from which projects tasks come because in this example the WBS numbers repeat in each project. Some organizations embed a project identifier at the beginning of all WBS numbers. That can be useful during project execution.

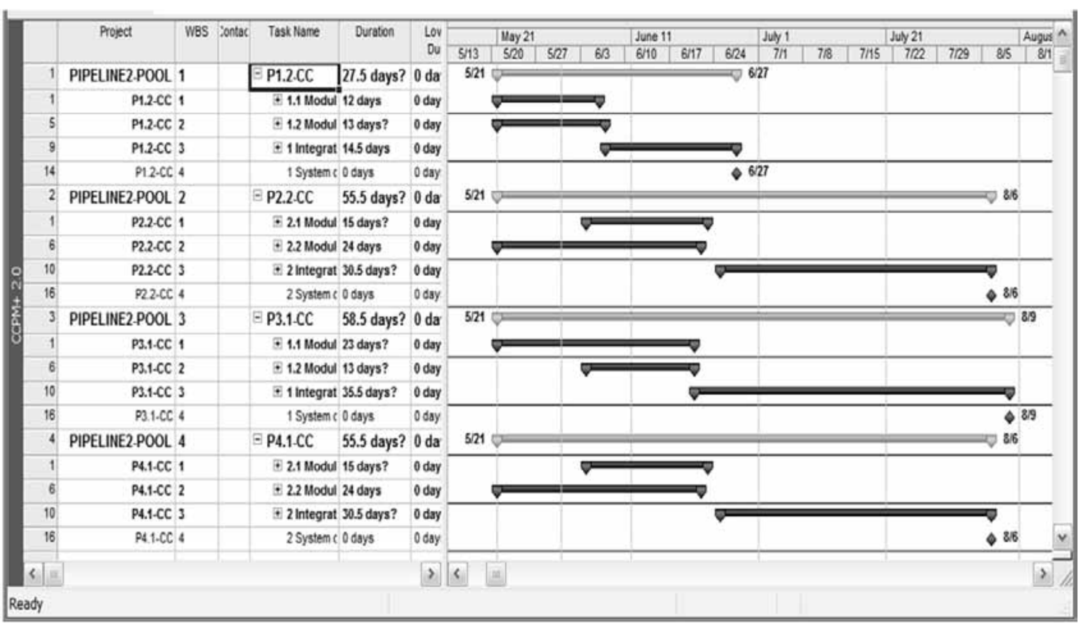


Figure 8.11 The initial Pipeline of four projects inserted into a single master project file.

8.6.2 Select the Drum Resource

The drum resource is the constraint to completing multiple projects sooner. You will use it to Pipeline the projects. It should be the most loaded (or overloaded) resource that is not easy to elevate. If you can easily elevate a resource to avoid it being overloaded, you should do so.

The example case has two highly loaded resources: designer and technician. Either could serve as the drum resource. I chose the designer resource as the drum for this pipeline.

Note that in most cases the supply of the drum resource will be more than one resource of that type (e.g., you may have four designers, or 400%, as in this example), but you may only be able to use a lesser number on a given project. For example, the projects may be at different locations, with less than the full number of a given resource available. In this example, the most applied on one task was at 200%, or two resources.

Figure 8.12 illustrates the Pipeline and the total demand for the Designer resource. I had to adjust the display to show the tasks in one project in order to show the resource to be selected to expand one project to show the resource to be selected for the graph below it. All of the projects share a common resource pool so that the resource demand chart illustrates the demand of the designer resource for the sum of the four projects. Although you cannot see the colors in this illustration, the area above the horizontal line at 400% on the resource graph shows in red; the resource is overloaded. The resource supply line shows at 400%, or four designer resources, are all that are available. For this example, let's assume there are six actual designer resources, so this line would allow a capacity constraint buffer of 50% of the resources scheduled or 33% of the designer resource pool. Note

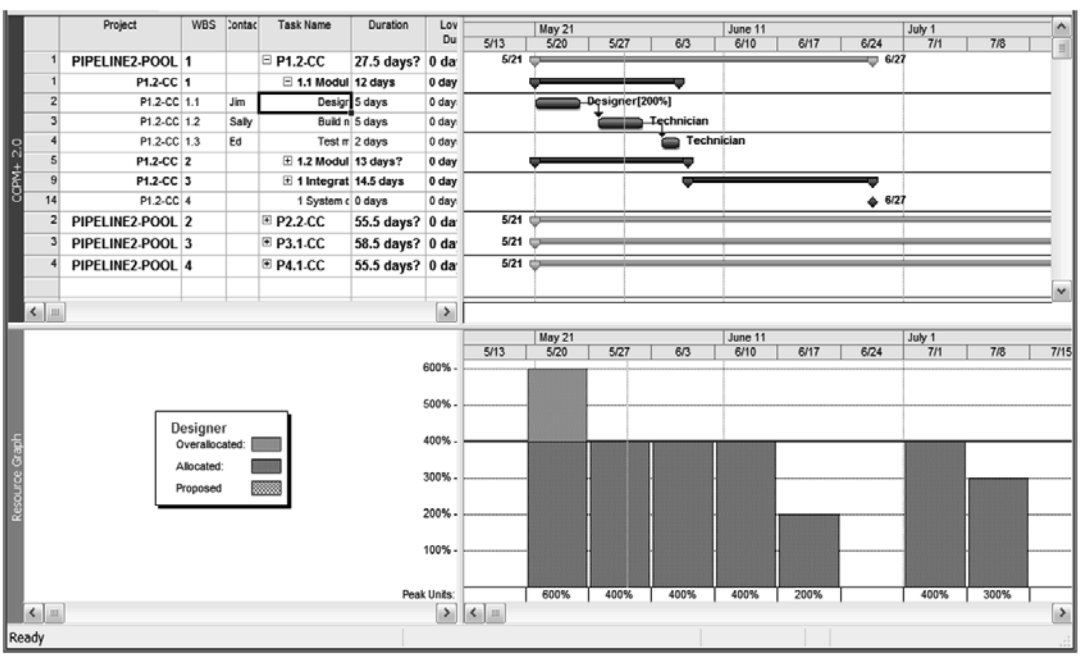


Figure 8.12 Pipeline and resource graph illustrating designer resource overload.

that the designer resource is loaded all the way to 600% (i.e., the resource demand peaks at 600%).

8.6.3 Decide on the Capacity Constraint Buffer

If you have only a few projects or only one unit of the drum resource, you can conceptualize Pipelining as placing a buffer between the use of the drum resource in one project and the use in the next sequence in the priority illustrated above for the simpler example. In that case you delayed the lower priority project a certain time relative to the preceding project. That time removed the direct conflict for the resource and added time for the capacity constraint buffer.

For larger projects and resource pools, you might conceptualize the capacity constraint buffer as preventing the demand for the pool of resources from rising above the supply for an extended period of time. Sequencing projects by delaying the start of some projects serves to lower the demand for the resource. Although you still delay the project, in time the measure becomes the relative demand for the resource as compared to the supply. This example uses this method.

For this example, only six designer resources (600%) are available and I have to allow for a capacity constraint buffer. Thus, it will be necessary to delay projects to create a feasible multiproject plan that does not overload the designer resource. For this example, I will set the capacity constraint buffer at 50% of the designer pool and thus seek to contain the designer resource below 300%.

8.6.4 Pipeline to the Drum Resource

Pipeline by adjusting the start dates for lower-priority projects later in time. The manual approach is a trial-and-error process checking the impact on the resource graph for the drum resource as you proceed. You should proceed a project at a time to pipeline manually. If you inserted the projects into the Pipeline file in priority order from top to bottom, you can follow that while delaying projects to achieve a satisfactory capacity constraint buffer.

Delaying the projects yields the summary result below, and the detail for the designer resource below that. This is a satisfactory Pipeline (see Figure 8.13).

Does it meet your customer needs? If so, proceed with it. If not, you must look for alternatives, usually to accelerate the project schedules.

With the various choices I made, the load for the drum resource is 200% or less for half the time you might sometimes have up to 600% of that resource available. That may be necessary if you anticipate frequent interruptions of project work. If those interruptions do not occur, you can complete the projects faster than plan. If you do not expect interruptions, you should ask, “Is this too conservative?” You may want to try a what-if analysis by leveling the pipeline to 400% for the drum resource.

Temporary overloads of the capacity constraint buffered resource supply, within the overall resource’s physical availability, can be acceptable because Pipelining used a capacity constraint buffer to set the available resource, and there may be extra capacity immediately before and after the peak so delays would be small. A detailed examination may show that the overload is only for a small fraction of the

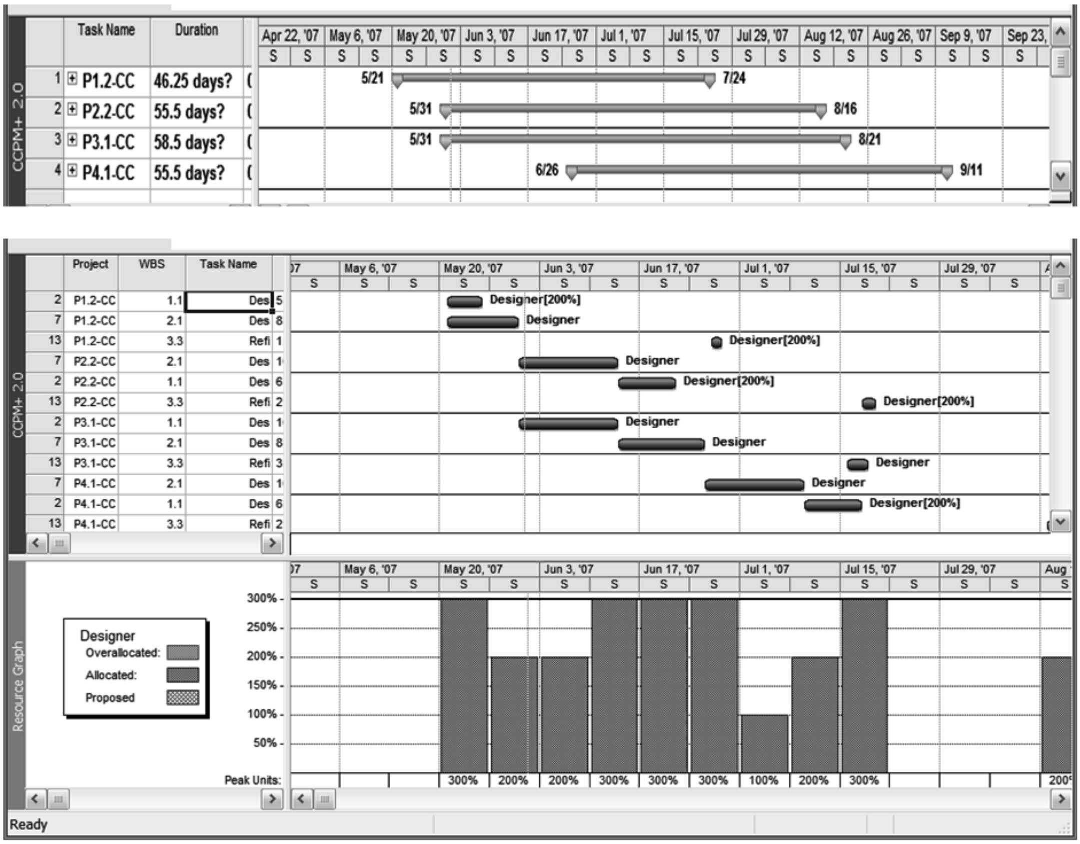


Figure 8.13 A satisfactory Pipeline solution shows no overload relative to the buffered supply of resources.

time shown. However, note in the case above that the resource is loaded to 300% for 3 consecutive weeks, so reducing the buffer may not be advisable.

8.7 Practical Pipelining Methods

Organizations use different ways to Pipeline for overall WIP control (demand leveling). Simple methods are fine. Some organizations simply limit the number of active projects and only start a new project when a project finishes. They pull new projects into the system as projects complete. If your projects are of relatively short duration and similar in size, duration, and drum resource demand, this might work for you. Other organizations use an Excel spreadsheet and simple templates for projects to Pipeline. The templates do not show the overall project schedule detail but just the overall length of the project and the number of drum resources demanded placed at the proper time location along the project line. Different templates accommodate different types of projects. This approach is similar to the "heijunka box" used to level demand in production by placing production orders into physical boxes to limit the WIP and level the workflow. Other organizations apply sophisticated multiproject CCPM software.

8.8 Frequently Asked Multiproject Questions

1. *Our project mix changes over time. How do we identify the drum?* It is a good idea to check the drum resource from time to time because it can change. When it does, it is time to reanalyze the pipeline. 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 that 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: behave to double throughput or do not. Once they see the results many management teams are able to do much better at this than they thought. After all, when the system makes more money, people's jobs are protected and often they make more money too.

8.9 Summary

This chapter describes Rule 3: Pipelining. It starts by clarifying that the critical chain for a single project is usually not the constraint for an enterprise performing multiple projects. You must identify the multiproject constraint and go through the focusing steps to adapt the CCPM process for multiple projects. When you identify the multiproject constraint and use it to schedule projects, it becomes the drum for your organization. The following list summarizes the key points for CCPM multiproject planning:

- Pipelining (rule 3) is the process of organization demand-leveling for projects.
- Pipelining maintains the WIP of projects to a level that maximizes throughput for the organization and minimizes the waste of excess demand and demand fluctuations.
- 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 maximizes throughput and minimizes cycle time by preventing excessive queuing delay.

- The capacity-constraint buffer should be in the range of 50% of the constraint-resource capacity.
- Individual project Critical Chain plans operate to the project start times developed from the drum schedule, including the capacity-constraint buffer.
- Management must introduce new projects to the system through the drum schedule by first assigning their priority relative to ongoing projects and 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.

References

Leach, L. (2005). "Applying the Theory of Constraints for Project Portfolio Management." In Levine, H. *Project Portfolio Management.* San Francisco, CA: Jossey-Bass.

Ohno, T. (1988). *Toyota Production System.* Portland, OR: Productivity Press.

Pascal, D. (2007). *Lean Production Simplified.* New York: Productivity Press.

Project Management Institute (PMI). 2013. *A Guide to Project Management Body of Knowledge, Fifth Edition.* Upper Darby, PA: Project Management Institute

CHAPTER 9

Execution

This chapter presents key CCPM execution actions in terms of the various project roles. It describes two types of measurements necessary to support execution: operational measures and performance measures. Task Managers 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. Performance measures can relate to results achieved or the behaviors believed to cause those results.

In the following list of what effective measures must do, Dr. Joseph Juran (1988) identified the first six items. He seems to have missed the seventh. Section 5.3 describes the background for exploiting the plan using buffer management and notes that Dr. Eliyahu Goldratt (1990) 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. The measurement and control system reinforces Rule 1: Focus through Rule 2: Buffer by applying buffer management and also must reinforce Rule 3: Pipeline. 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. Provide positive, immediate, and certain (PIC) feedback for desired behaviors.
10. Be simple.

The Lean precepts of focusing on flow, pull, and using visual controls are also important considerations to designing an effective measurement-and-control system for CCPM. The system must eliminate the muda (waste) of complexity. In his book *Gemba Walks*, James Womack (2011) said that you have to eliminate mura (unevenness of load, starvation and overload of the constraint) and muri (overburden, too much WIP) before any efforts to eliminate muda will be long-lasting.

The CCPM process does this at the system level through the critical chain and Pipelining and Kanban does it at the local level.

Daniels and Daniels (2006) provided a useful perspective on how to design measures to elicit desired human behavior. They clarify that you should have both results measures and behavior feedback measures that cause people to behave as needed to create the desired results. Behavior measures need to provide positive, immediate, and certain reinforcement to be effective at increasing desired behavior. When a resource completes a task using the desired behaviors and moves the task to the complete column on a Kaban board, a Task Manager saying, "thank you for focusing on and completing that task," meets those requirements. Interestingly reinforcements that are variably spaced in time are more effective than reinforcements at a fixed interval.

The goal of all projects in all organizations within a company or institution must relate to the company or institutional goal. For a profit-making company, this goal is to make money now and in the future. As noted in Chapter 2, 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.

This chapter starts with defining the various roles important for CCPM execution because it helps to think of measures appropriate to each role. I then describe project schedule and cost buffer management, the latter in more detail than I have covered it to this point, including responding to buffer signals. I then cover a number of other measurements for control including quality, Kanban, and measuring milestones. I end the chapter with a discussion of change control and some frequently asked questions.

9.1 Project Roles

This chapter addresses measurement and control to the plan for five different roles: the Task Manager, the Project Manager, the Resource Manager, the Master Scheduler, and Senior Management.

9.1.1 Task Manager Role

Task Managers maintain the workflow. The Task Manager needs to know how to guide his team as to which task to work on next so that they can pull the appropriate tasks for focused execution. The challenge is to work on the tasks that will move projects toward completion as soon as possible while at the same time handling all of the nonproject work assigned to the group.

The Task Manager is responsible and accountable for task completion. The Task Manager most often directly supervises the task performing resource, although sometimes he or she may also be the performing resource. The most important

Task Manager behavior is to positively reinforce desired behaviors by the team. Task Managers can manage more than one task at a time, frequently both planning for future tasks and leading execution on present tasks.

Task Managers are responsible for the following key project functions:

- Task planning:
  - Identify all of the tasks necessary to complete the project deliverables.
  - Identify the relationships among the tasks.
  - Estimate the duration for the tasks.
  - Identify the resources necessary to perform the task.
- Task execution:
  - Shield resources from interruptions.
  - Authorize the resources to start work on a task.
  - Report task status upon start, completion, and at identified intervals.
  - Help resources get what they need to not delay tasks.
  - Escalate tasks for help when unsticking tasks requires action not within the direct control of the Task Manager.
  - Support buffer recovery.

Due to their vital role, Task Managers must be accessible at all times or designate an alternate. Task Managers most often are first-level supervisors.

Task Managers must estimate tasks to effectively apply resources to complete all of the projects in the portfolio as quickly as possible. The duration estimate should provide a 50% chance of completing the task within the duration estimate assuming the performing resource that determines the task duration works in increments of 100% of their time. For the first Critical Chain project, determine this estimate by decreasing the duration estimate previously used, keeping the work the same (i.e., person-hours), to achieve at least 100% resource loading on the task, and then cut that duration in half.

Task Managers enable resources to pull tasks for execution and then focus on helping the task performers complete the tasks as soon as possible. Many Task Managers have short daily stand-up meetings with their task performers at their Kanban board to pull new tasks from the queue and provide direct help or escalation for help on tasks that are stuck. The Task Manager must lead the group to avoid extraneous discussions during such meetings or they will drag on. There is a powerful human need to talk about what they accomplished. A Task Manager just needs to know where and when they need help. Wherever possible, the team members should move the cards before the meeting so they do not get triggered into discussing what they accomplished. Issues should be raised during the meeting and actions assigned but not necessarily resolved unless the resolution is very quick.

Task Managers report task status to the Project Manager. They provide the actual start date when tasks start, the actual finish date when tasks finish, and an estimate of remaining duration for in-progress tasks at the standardized status frequency, most often once per week on a particular day. With a properly designed Kanban board, the Task Manager can read this information off the board. One approach is to include an estimate of remaining duration on the top of each column on the board.

Task Managers should accumulate actual task performance data for future planning and execution, including a control chart of the ratio of actual task performance duration to estimated duration and track appropriate Kanban measures such as WIP.

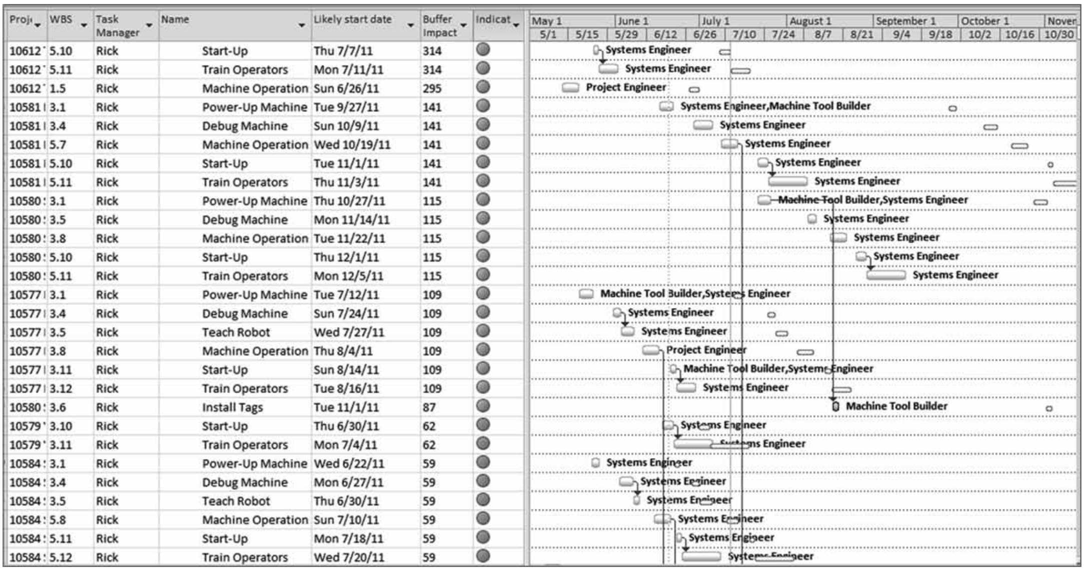
W. Herroelen, R. Leus, and E. Demeulemeester (2002) criticized CCPM for not "rescheduling" projects dynamically: "Opportunities for speeding the remaining part of the ongoing project may be exploited by rearranging the schedule" (p. 57). They did not understand that CCPM does not schedule task dates at all and that by performing task assignments dynamically it achieves the same end as continuous rescheduling, accelerated project completion, without the waste of continuous rescheduling. They did not know about using Project Kanban at the work team level to prioritize the flow of work locally.

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 makes a good case for following the CCPM approach.

Rule 1 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 (or allow the resources to pull from the Kanban queue) the next project 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.

CCPM greatly facilitates this decision by providing the Task Manager with a prioritized list of tasks in the order in which they should start. Most CCPM software provides such prioritized task lists filtered by Task Manager or you can automate this process with some non-CCPM software. For example, two of my clients have used Primavera by instituting a script to generate the prioritized task 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 (Advanced Projects, 2013) takes a graphical approach, showing the tasks that are ready to work in priority order (Figure 9.1). Other software automates the process for multiple projects providing the Task Manager a prioritized list of tasks designed to facilitate rapid task-statusing and communication. Note that the information is presented for multiple projects (in this case designated by a number) but filtered for one Task Manager and then sorted by buffer impact. The round indicators show the color associated with the amount of buffer penetration. (Although you cannot see the colors here, the top ones are



Proj	WBS	Task Manager	Name	Likely start date	Buffer Impact	Indicat
10612	5.10	Rick	Start-Up	Thu 7/7/11	314	
10612	5.11	Rick	Train Operators	Mon 7/11/11	314	
10612	1.5	Rick	Machine Operation	Sun 6/26/11	295	
10581	3.1	Rick	Power-Up Machine	Tue 9/27/11	141	
10581	3.4	Rick	Debug Machine	Sun 10/9/11	141	
10581	5.7	Rick	Machine Operation	Wed 10/19/11	141	
10581	5.10	Rick	Start-Up	Tue 11/1/11	141	
10581	5.11	Rick	Train Operators	Thu 11/3/11	141	
10580	3.1	Rick	Power-Up Machine	Thu 10/27/11	115	
10580	3.5	Rick	Debug Machine	Mon 11/14/11	115	
10580	3.8	Rick	Machine Operation	Tue 11/22/11	115	
10580	5.10	Rick	Start-Up	Thu 12/1/11	115	
10580	5.11	Rick	Train Operators	Mon 12/5/11	115	
10577	3.1	Rick	Power-Up Machine	Tue 7/12/11	109	
10577	3.4	Rick	Debug Machine	Sun 7/24/11	109	
10577	3.5	Rick	Teach Robot	Wed 7/27/11	109	
10577	3.8	Rick	Machine Operation	Thu 8/4/11	109	
10577	3.11	Rick	Start-Up	Sun 8/14/11	109	
10577	3.12	Rick	Train Operators	Tue 8/16/11	109	
10580	3.6	Rick	Install Tags	Tue 11/1/11	87	
10579	3.10	Rick	Start-Up	Thu 6/30/11	62	
10579	3.11	Rick	Train Operators	Mon 7/4/11	62	
10584	3.1	Rick	Power-Up Machine	Wed 6/22/11	59	
10584	3.4	Rick	Debug Machine	Mon 6/27/11	59	
10584	3.5	Rick	Teach Robot	Thu 6/30/11	59	
10584	5.8	Rick	Machine Operation	Sun 7/10/11	59	
10584	5.11	Rick	Start-Up	Mon 7/18/11	59	
10584	5.12	Rick	Train Operators	Wed 7/20/11	59	

Figure 9.1 CCPM+ Software shows a prioritized, filtered view for Task Managers to decide on which project tasks to focus.

red and those with buffer penetration of 62 and below are green on this example.) The Task Manager should focus on the red tasks first and yellow tasks second (if any), allowing green tasks to sit in the queue if necessary. If multiple red tasks exceed available resources, the Task Manager should use other information at their disposal to decide priority, for example, how much of the critical chain is complete (usually better to first perform tasks on projects that are nearly done) or the rate of buffer use (usually better to help on projects with a high rate of buffer use).

Note that if all tasks are green or if many tasks can sit in the queue for extended periods, that is an indication that task duration estimates may be inflated. Likewise, if all or many tasks are red, it would be an indication that task duration estimates are too short or buffers are too small. The Task Manager should address such overall trends in the weekly team meetings and monthly improvement meetings when necessary.

The Task Manager plays the key role during project execution by making the whole CCPM system work: actively leading task execution, resolving issues, 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 must be trained and reinforced 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 Manager 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.

Task Managers are key to applying the Kanban process. They lead the team. They work with all customers of the team to pull tasks into the Kanban queue. This is where they assign relative priority to project and nonproject work tasks subject to the overall WIP limits that the team works to. They reinforce the operating rules of Kanban.

Measures that might help reinforce desired behavior by Task Managers include:

- Performing resource positive reinforcements (number per day per resource);
- Percentage of tasks complete in less duration than the estimate;
- Percentage of tasks complete above the estimated duration;
- Task output meets quality needs (percentage or number of defects);
- Task status reported when started, when finished, and weekly (percentage);
- Task recovery actions taken on time (percentage).

Reinforcement must only occur for proper behaviors and not for results achieved by undesired behaviors. Be careful reinforcing results unless you know the behaviors that produced those results. Otherwise, you might inadvertently reinforce the wrong behaviors, and you are encouraging more of the same in the future. The most important reinforcements that Task Managers can provide their team are verbal and written acknowledgment for following Rule 1 on their tasks and asking for help as soon as a task gets stuck.

9.1.2 Project Manager Role

The Project Manager controls the project value stream. During execution, the Project Manager's primary question is, "When should I take action to recover buffer?" The 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.

Project Managers are responsible and accountable for all aspects of project planning and execution. The most important Project Manager behaviors are to create effective Project Plans and to lead execution to that plan. Project management requires specific knowledge and skills. Project Managers of larger projects should be certified as Project Management Professionals (PMPs), and for smaller projects should at least have or be seeking certification as a Certified Associate in Project Management (CAPM) from the Project Management Institute.

Project Managers are responsible for the following key project functions:

- Project planning:
  - Develop all elements of a Project Plan as necessary and appropriate for the project.
  - Develop the Critical Chain schedule for the project, and cause it to be pipelined.

- Hold a project kick-off meeting with all project stakeholders to ensure buy-in to the plan.
- Project execution:
  - Authorize Task Managers to start execution in accordance with prioritized task lists.
  - Ensure effective task statusing and execution.
  - Lead weekly buffer management meetings to ensure project flow.
  - Manage project changes, issues, and actions, escalating to Senior Management as necessary to achieve the project goal.
  - Reinforce effective behaviors on the part of all project stakeholders.

Due to their vital role, Project Managers must be accessible at all times or designate an alternate.

Project Managers develop the necessary elements of the Project Plan, including the critical chain schedule, starting with the customer's statement of work, a Work Breakdown Structure (WBS), and project planning assumptions. They develop the critical chain schedule to the WBS and ensure that customer need dates are met as outputs from the schedule. They communicate the Project Plan (also known as full kit) to all project stakeholders before starting work on the project.

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 (someday, 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 compared 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 9.2.1) 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.

Buffer reporting relies on realistic estimates of how many days remain to complete a task or RDU (remaining duration), expressed in working days. There is often a tendency to report a project task as on schedule until the task due date arrives. With the CCPM 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 for tasks 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 percentage of buffer consumed versus the percentage of critical chain accomplished as illustrated by Figure 9.2. 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 (1986) noted that the trend information is even more important in such cases.

Figure 9.3 illustrates the project chain view. It is a view for the incomplete tasks in single project over the next periods of time. This project chain view sorts the tasks by the amount of buffer penetration they are causing. The sort causes tasks to line up in chains with the same buffer impact. The chain of tasks with the same buffer impact is the set of opportunities for recovering buffer.

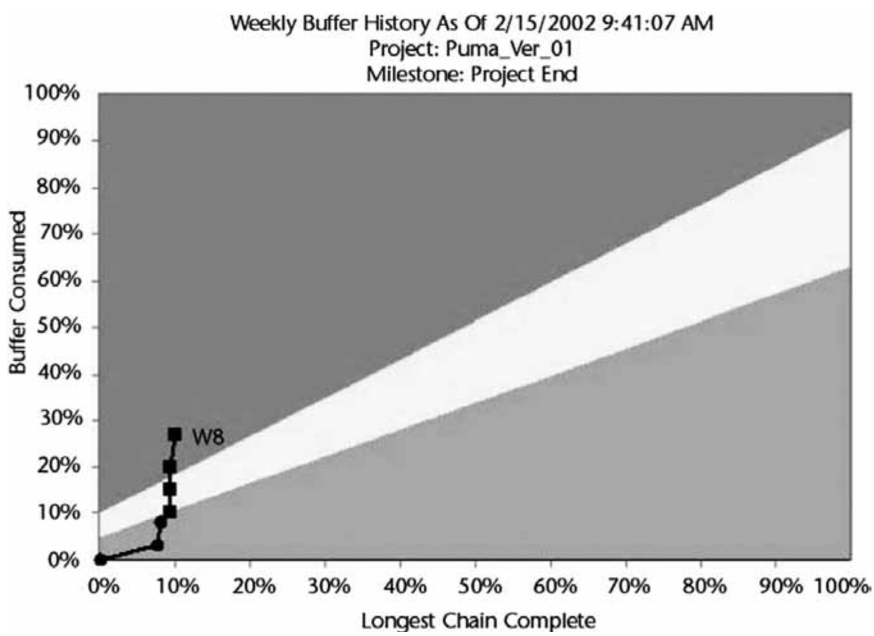


Figure 9.2 Measuring the trend of buffer penetration improves the early-warning aspect of buffer management.

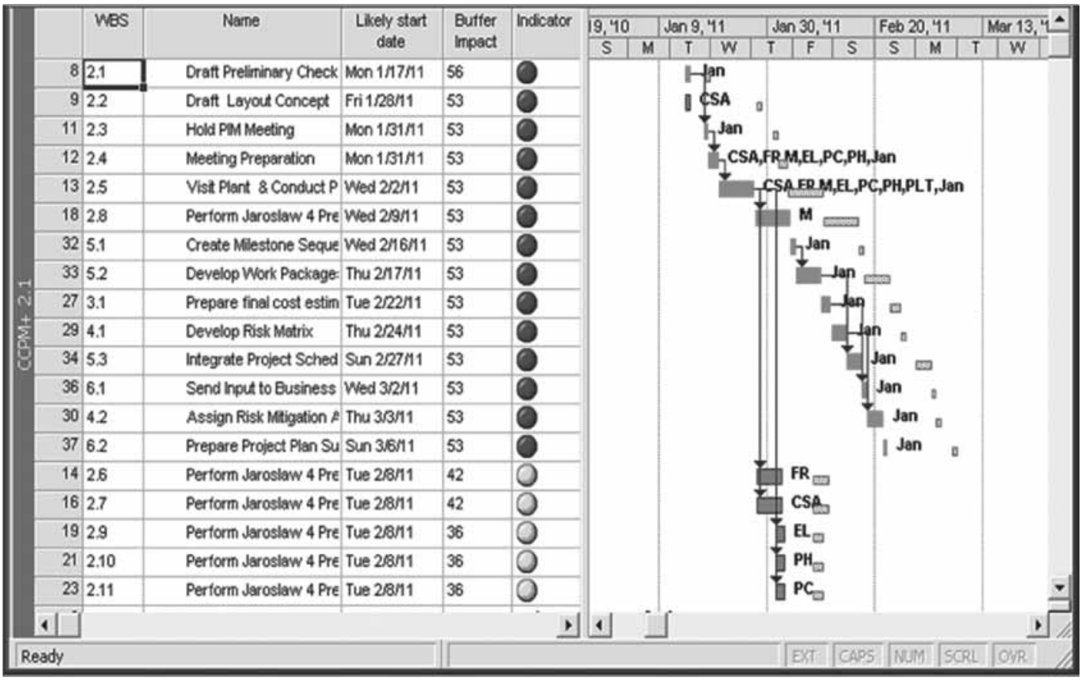


Figure 9.3 The project chain view aids project teams in planning buffer recovery.

The Project Manager can also see if there is a chain with nearly the same buffer penetration as the most penetrating chain. If so, it may require buffer recovery as well. The project chain view is the most important view to use during project weekly meetings.

Measures that might reinforce desired CCPM Project Manager behavior include:

- Task Manager positive reinforcements for focused task behavior (number per day);
- Tasks started using task priority (number of demanded red tasks not started);
- Task status reported (percentage of tasks);
- Task output meets quality need (number or percentage of defects);
- Projects perform below red regions of fever chart (percentage of status updates in the red);
- Buffer recovery actions taken on time (percentage).

Of course, the standard results measures for projects apply as well: measures that relate to scope, schedule, and cost performance.

9.1.3 Resource Manager Role

Resource Managers provide the resources necessary to deliver quality work on project tasks. The most important Resource Manager behavior is to positively reinforce desired behaviors by supervisors. Section 9.1.1 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.

Resource Managers are responsible for the following key project functions:

- Support of portfolio and project planning:
  - Identify all of the available resources in the planning resource pool.
  - Staff and train to supply the resources and Task Managers required by the project pipeline, including protective capacity.
- Execute: allocate resources to Task Managers for project work.
- Measure resource performance (results and behavior):
  - Reinforce desired Task Manager, supervisor, and resource behavior to minimize multitasking and deliver quality task results.
  - Promptly resolve tasks escalated by Task Managers.
  - When requested by Project Managers, support buffer recovery.
  - Lead continuous improvement.
  - Measure and improve the performance of Task Managers.
  - Improve the work processes for task delivery.

Measures that might reinforce Resource Managers to exhibit the desired CCPM behaviors include:

- Supervisor behaviors reinforced (times per day);
- Percentage of escalated tasks cleared in one day;
- Task outputs meet quality needs (number of defects);
- Multitasking (number of started but incomplete tasks divided by number of available resources);
- Number of tasks per resource per week stopped without completion.

Senior Management needs to reinforce Resource Manager behavior by providing positive, immediate, and certain feedback to Resource Managers who properly execute the key functions listed above in order to maintain and improve on their measures.

Resource Managers must be accountable for variation in resource availability relative to their demand by the projects (e.g., vacations, sick time, terminations, and problems that might cause excessive demand on particular resource skills). They need to pay particular attention to the drum resource and to resources that sometimes become capacity constraints due to statistical fluctuations. They need to engage in providing back-up resources for such contingencies.

9.1.4 Master Scheduler Role

In many organizations the Master Scheduler fills a twofold role: the scheduling subject matter expert (SME) and pipeline creation and maintenance. The Master Scheduler acts as the process owner for project scheduling and for developing and maintaining the project pipeline. He or she also acts as the senior subject matter expert for scheduling across the organization. The PMI Project Manager competency guide describes many of the competencies that the Master Scheduler must possess (PMI, 2007).

When starting CCPM most organizations do not have anyone doing the pipelining role. Master Schedulers should be subject matter experts in project and program management. They should be certified as a PMI Scheduling Professionals (PMI-SP), Project Management Professional (PMP), or equivalent. They must be skilled in the application of the organization's scheduling tools and stay abreast of knowledge in the field of project and program schedule management.

Master Schedulers must have written and oral communication capabilities, and be proficient with common office software such as e-mail, word processing, spreadsheets, and graphics. They should also have knowledge of how to establish and operate collaboration tools, such as SharePoint sites or Wikis.

As the scheduling process owner, the Master Scheduler functions include the following schedule process items:

- Process design;
- User training;
- Maintenance;
- Operation;
- Improvement.

As the scheduling subject matter expert, the Master Scheduler functions include for all of the schedulers in the organization:

- Acquiring;
- Training;
- Providing input for performance development.

The Master Scheduler also operates the pipelining function, including:

- Maintain the project priority list.
- Add, delete, or modify projects in the pipeline.
- Perform what-if analyses of enterprise scheduling alternatives.
- Establish the start date for projects.

9.1.5 Senior Management Role

Senior Managers design, lead, and improve the organization system to deliver on all customer needs. The most important Senior Manager behaviors are to design an effective system, control WIP, positively reinforce desired behaviors by middle managers, and coach process improvement.

Senior Managers are responsible for the following key functions:

- Develop and continuously improve organization-level work process:
  - Set the vision for the organization.
  - All elements of the "Seven S" model of business system design.
  - Implement the three rules.
  - Design and implement the work processes necessary to deliver on the vision and the three rules.
- Lead execution:
  - Measure system performance.
  - Make decisions and take actions necessary to satisfy customers.
- Prioritize all work:
  - Control Work in Progress (WIP) to ensure maximum system throughput and quality.
  - Reinforce desired middle manager behavior to deliver quality project results.
  - Promptly resolve escalated tasks.
  - Support buffer recovery.
- Continuously improve work processes:
  - Measure and improve project delivery system performance.

Controlling WIP enables improving due date performance, project throughput, and project quality by enabling resources to focus on the right tasks. Only Senior Management can control WIP because work requests can come from many sources outside and inside the organization. In the beginning, the system likely has much more WIP than appropriate to function effectively. The excess WIP causes dysfunctional behavior, such as multitasking and frequent priority switching, which leads to late delivery, cost overruns, low throughput, and poor quality. Senior Management must take decisive action to reduce this WIP or system results will not improve. Whether they do this by "freezing" projects or Pipelining, they must initially stop work on some projects and/or nonproject work to complete all projects sooner and establish new organizational norms of project delivery. At the beginning, Senior Management needs to educate the internal and external customers on what to expect as the initial pass at WIP management suspends work on a significant number of projects that have already started. Then they must control input to match system delivery capacity. Unfortunately, many Senior Managers have great difficulty focusing on and performing this role.

Senior Managers lead ongoing project estimating and delivery process improvement. They measure actual project performance data for future planning and execution and coach the use of Six Sigma and Lean tools such as Pareto charts and control charts to identify effective improvement opportunities. They lead the overall process improvement.

Measures that might reinforce Senior Management to perform to the desired CCPM behaviors include:

- Middle management CCPM behavior reinforced (times per day);
- Percentage of escalated tasks cleared in one day;
- Projects always work to an approved pipeline (percentage of compliance);
- Project management decisions rely on statused schedules (number of defects per quarter);
- Project WIP controlled to deliver on all client commitments (number of projects in progress);
- Projects complete with less than 100% buffer penetration (percentage of defects).

More Senior Management must provide positive, immediate, and certain reinforcement when senior managers that report to them exhibit the desired behaviors as shown by both behavior and results measures.

9.2 Schedule Buffer Management

The measurement system for CCPM 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 5.3 recommended explicit action levels for the Project Manager's primary decision: when and where to take action to complete the project successfully. The project buffer is the most important monitoring tool.

9.2.1 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 most meeting participants' time by discussing each task on the schedule. Any one task usually interests at most a couple of people in the meeting. 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, and so forth).

Figure 9.4 illustrates an example format for a weekly project meeting buffer recovery plan. These can be created in the buffer meeting if the solution assignment is obvious or they can be created and distributed after the buffer management meeting by the person assigned the action to lead buffer recovery. Note that the format can act as the meeting minutes by providing space for new issues, project changes, and a periodic look at risk management.

Project Name ________________

PM: ________________

Contact Number: __________ Email: ______________

Project WEB: ________________

Buffer Status Date: ____________

% PB Used: ____________

% CC Cpl. : __________

Buffer Status:
- ☐ Act
- ☐ Plan
- ☐ Watch

Actions to Unstick Tasks:

#	WBS	Task Mgr.	Due	Action

Actions to Recover Buffer:

#	WBS	Task Mgr.	Due	Action

New Issues, Actions, or Decisions:

#	WBS	Task Mgr.	Due	Action

Project Changes:

NEW: ________ Approved: ________ Pending: ________

Risk Management Review:

Risk #	Risk Mgr.	Due	Action

Figure 9.4 Example of a buffer recovery plan format for weekly meetings.

9.2.2 Task Manager Meetings

Task Managers lead three types of meetings:

1. Daily meetings with their team to identify issues with blocked tasks and facilitate pulling of tasks across the Kanban boards;
2. Weekly (or more frequent) meetings with their team's customers to promote items from the backlog onto the Kanban board weekly work queue;
3. Weekly meetings with their team to identify and move forward opportunities for improving the work process.

9.2.3 Senior Manager Project Meetings

Senior Managers lead three types of meetings focused on projects:

1. Meetings to set project priority for Pipelining and control overall project WIP;
2. Meetings to monitor project progress and determine where they need to take action to help on project flow;
3. Meetings to lead continuous improvement of the work processes.

The project WIP control and priority meetings should take place as needed but no less often than about half the average duration of projects in the Pipeline. So if projects are nominally a year-long, the project priority meeting should take place at least every six months. There may be a need for an ad hoc process to determine the priority for projects that enter the system between the overall pipeline priority meetings.

The meetings to monitor project progress and lead continuous improvement should be no less frequent than monthly but weekly is better.

9.2.4 The Buffer Report

Clients always want to know how their project is going. Project management sometimes wants to separate the clients 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 do not really know the answer to. (Everybody likes to help.)

You are probably well 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 projects because they are not timely. Control has too take place at the frequency of task completion: weekly or more often. Monthly meetings can be useful for project portfolio decisions (e.g., should we cancel this project?), for long-term resource-management decisions (e.g., should we hire more resources of a particular type?), and for planning major improvement initiatives, although once an improvement initiative starts, it needs to be controlled with the same timeliness as any project.

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 9.5 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. Figure 9.6 illustrates a popular way to display overall progress on a portfolio of projects: the multiproject fever chart.

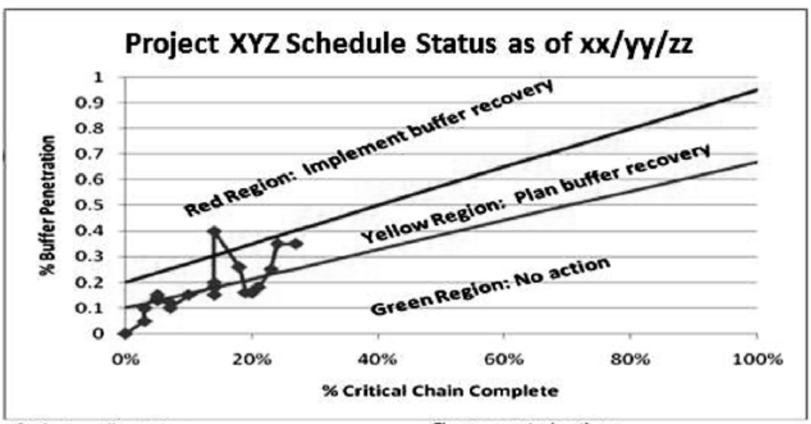
Management often overlooks the project team as the recipients of project reports. The project team rarely has the time to read thick reports and often does not have access to them. There is no excuse for failing to make the information

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 9.5 Example of project-status report that plots buffer trends.

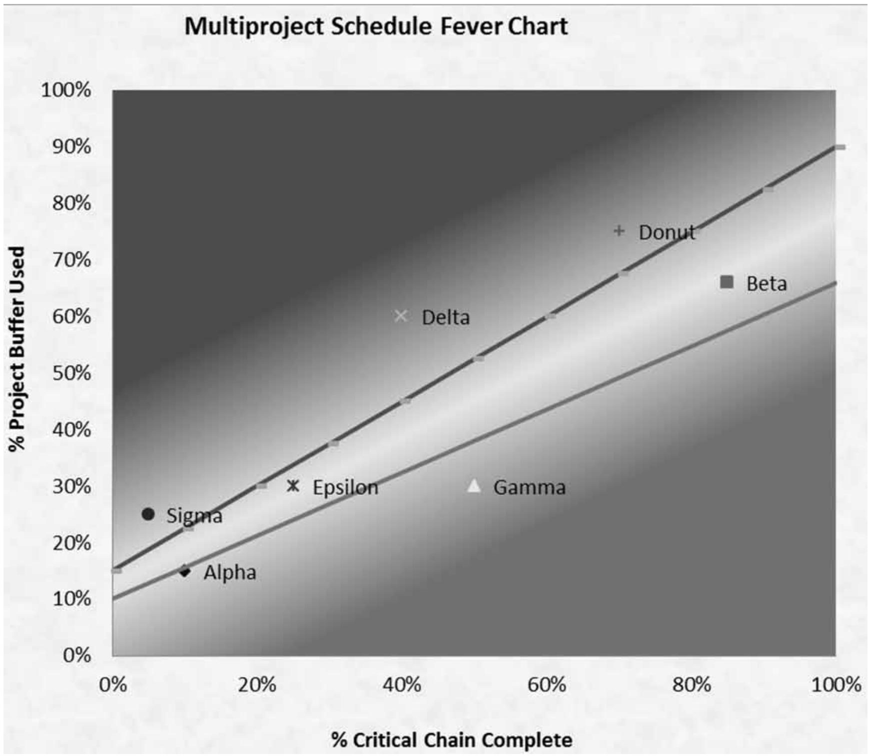


Figure 9.6 Example of schedule progress on a portfolio of projects using the fever chart.

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.

9.3 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 7.5 addressed 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 appropriate currency units or percent of buffer, 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.

9.3.1 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. 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.

9.3.2 Earned-Value Basics

Earned value defines three terms:

1. *Actual cost (AC):* This is simply how much you have spent to date on a task or the project; this is also called the ACWP (actual cost of work performed).
2. *Planned value (PV):* This is the estimated cost for a task or the sum of the estimated costs for tasks up to a point in the schedule; this is also called the BCWS (budgeted cost of work scheduled).
3. *Earned value (EV):* The estimated cost for completed activities. Never mind if the activity actually cost more or less than the budgeted amount. Earned value is a way of measuring progress against the schedule or cost plan; it is also called BCWP (budgeted cost of work performed).

Note that these definitions use the terminology adopted by the Project Management Institute (PMI). The four-letter acronym terms are the original terms and are still widely used outside the PMI community. I will use only the PMI terminology from here on.

The only new term here is EV: the earned value. The difference between cost to date on your project (AC) and the budgeted cost to date (PV) is the spending variance. Spending variance is made up of two parts: the cost variance (CV) and the schedule variance (SV). EV itself was the genius idea of the earned value process, allowing you to separate out how much of the spending variance is due to overrunning or underrunning the estimates and how much is due to being ahead-of or behind schedule.

9.3.3 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.

$$CV = EV - AC$$

This part of earned value can work hand in hand with CCPM, subject to some of the considerations addressed later on determining AC.

The cost-performance index (CPI) is used in earned value to judge the cost health of the project. A CPI greater than 1 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 1 is bad: less work is being accomplished than money spent.

$$CPI = EV/AC$$

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 EV. The AC is your actual project cost as of a given date.

9.3.4 The Problem

A problematic element of earned-value thinking is that it is subtly deterministic: it does not address variation or buffers. Figure 9.7 illustrates the overall project budget versus time, considering the project-schedule buffer and the 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 EV, 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

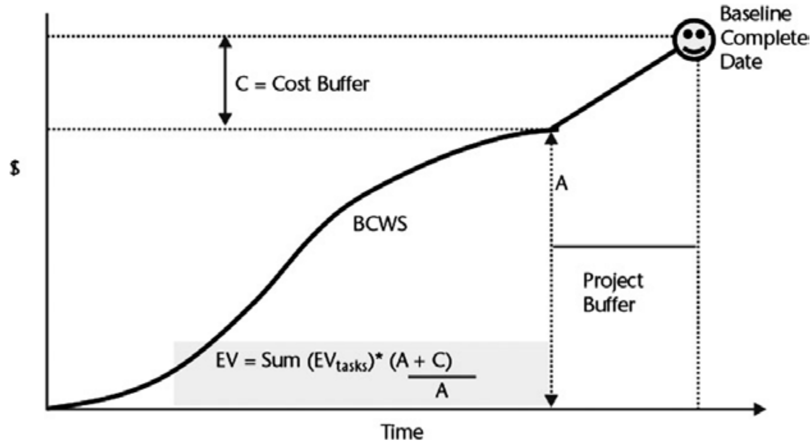


Figure 9.7 Project budget, showing schedule and cost buffer.

project buffer. Thus, one way to bring CCPM and earned value into line is to adjust the EV to account for the buffer as indicated in Figure 9.7. 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 1 to account for the cost buffer.

9.3.5 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 EV and AC for direct-labor costs on a daily basis, sometimes with a little lag. Strict TOC thinking identifies 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) cost mixing up labor cost with meaningless cost allocations to labor. However, 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 have trouble with achieving comparable values for material costs.

9.3.6 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 that 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 AC. 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.

9.3.7 Peaceful Coexistence of Buffer Reporting and Earned Value

You can manage the cost buffer the same way you manage the schedule buffer. Figure 9.8 illustrates the cost buffer fever chart, plotting the percent of cost buffer consumption versus the percentage of task budget completed. BAC equals budget at

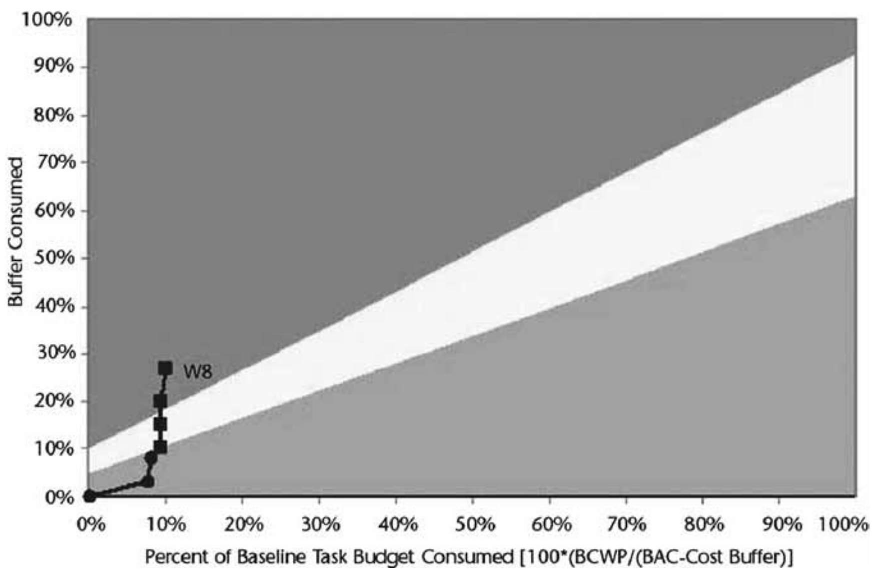


Figure 9.8 Cost buffer fever chart provides cost tracking and decision information.

completion for the project, including the cost buffer. The percentage of cost buffer consumed is the CV divided by the cost buffer, expressed as a percent. The percentage of the task budget complete is the earned value (EV) for completed work, as a percentage of the total EV for all project tasks. You can use any of the conventional earned value methods to accumulate EV for partially completed tasks, for example:

- No value until task is complete.
- 50% of value at task start, 50% at task completion.
- An estimated proportional amount.

As a relatively small number of tasks should be working at any time, it usually makes little difference which method you choose.

Figure 9.8 illustrates a cost buffer fever chart, similar to the schedule buffer fever chart.

9.3.8 The 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:

$$SV = EV - PV$$

$$SPI = EV/PV$$

The 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 of when you are going to be done. Unlike using CPI to predict the estimate at completion (EAC), you cannot use SPI to predict when the project will finish. By weighting cost, the schedule measures 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 0 and an SPI of 1. A project that takes twice as long, when complete, has an SV of 0 and an SPI of 1. These metrics have no relationship to the actual schedule. Use the CCPM buffer penetration metric for schedule measurement and control.

PMI (2005, p. 18) recognized this problem with the "S" measures of earned value as "measures of work scope, not time." I prefer to think of them as measures of what amount of the project scope has been completed rather than relating to schedule. Because all the project scope is done when the project completes, it is appropriate to have the scope variance go to zero whenever the project completes.

9.4 Responding to the Buffer Signals

9.4.1 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 multiple ways to reduce activity time: increase resources, change the strategy for doing the work to deliver the full scope, reduce the scope, or improve the process for the activity. Table 9.1 lists ideas to help reduce schedule-buffer penetration.

9.4.2 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 9.2 lists ideas to help reduce cost-buffer penetration.

9.4.3 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 overtime on such tasks.

Table 9.1 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 9.2 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	
Change strategy for performing the scope	Change strategy for performing the scope	Change strategy for performing the scope

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-control 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 schedule accordingly. Changes such as authorizing overtime or using contract labor should not require a change to your schedule.

9.4.4 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.

9.4.5 Schedule or Cost Buffer Exceeds 100%

If you have implemented 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.

9.5 Quality Measurement

Lewis Ireland (1991) described 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.

9.5.1 Basic Quality Measurements

Ohno (1988) noted, "In business, excess information must be suppressed" (p. 30). Six Sigma applies a range of tools, from simple charts to sophisticated statistical methods. It is easy to get lost in the detail of analysis. Be sure to first focus on the constraint and how to improve its throughput. Recall that experience with CCPM demonstrates substantial quality improvement occurs due to reducing multitasking and WIP. You want to ensure measurements reinforce the behaviors necessary to achieve these results.

Two of the most basic improvement tools emphasized by W. Edwards Deming are process flow charts and run charts. Process flow charts are essential to understanding a process for improvement, and to help identify the constraint. Process flow charts (see Figure 9.9) apply at both the overall project delivery system level, and at the levels of the individual project tasks and operational control mechanisms (e.g., the change control process). Run charts are the first basic step to collecting time series data for subsequent analysis and control. The most basic run charts for LPM are project schedule performance (schedule buffer penetration at project completion), cost performance (cost buffer penetration at project completion), and an appropriate quality measure (e.g., number of defects). Run charts are easily converted to more useful control charts as processes are brought into statistical control.

Seven basic tools of quality assurance have been used in wide application, to include:

1. *Cause-and-effect diagram (also known as Fishbone or Ishikawa diagram):* determines the likely cause of process output changes.
2. *Pareto chart:* categorical histogram, to identify the most significant contributors to an effect. Use in LPM to identify the primary contributors to buffer penetration.
3. *Checksheet:* used to assure quality (e.g., list of the inputs necessary for a project task).

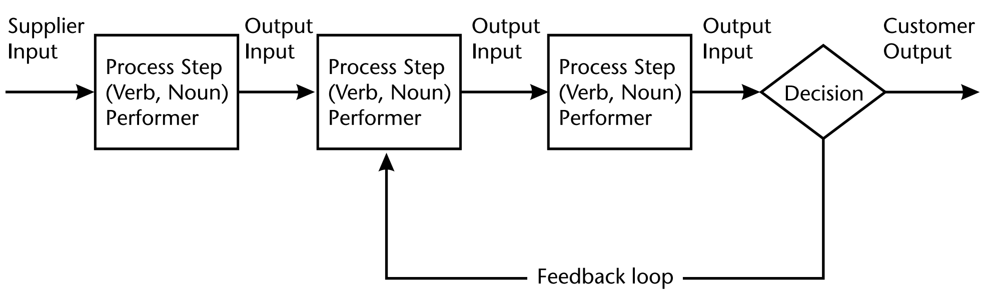


Figure 9.9 Process management begins with a process flow chart or process map. The process map links process steps. Process steps include supplier, input, process, output, and customer for the process result. Links between steps identify the supplier (predecessor step) and customer for the process (successor step). Process flow charts show decisions and feedback loops.

4. *Histogram:* charts the distribution of data from a process (e.g., the actual duration of project tasks, or the ratio of actual to planned duration).
5. *Scatter diagram:* x-y plot to determine the relationship between two variables (e.g., actual task duration versus estimated task duration).
6. *Process behavior charts (see next section):* used to control both the project delivery process and the processes that comprise tasks. The most basic ones in LPM are the ones for schedule and cost buffer penetration at project completion.
7. Various graphs.

Details of the sophisticated statistical tools are beyond the scope of this text. My purpose is to focus on the tools necessary to improve the project delivery system. Improvements to the project delivery system focus on the throughput of the system, where throughput implicitly includes quality products that satisfy customer needs.

9.5.2 Process Behavior Charts

Process behavior charts provide a powerful method to present data for management decisions during execution of CCPM projects and to guide improving the project delivery process. Wheeler (2000) proposed the name process behavior charts as a better descriptor of the quality control chart invented by Shewhart (1986). He also redefined an "out-of-control process" as a "unpredictable process" and an "in-control process" as a "predictable process." Whichever nomenclature you prefer, process behavior charts are a tool to guide management actions on a process.

Management of variation becomes possible when system variation is stable. Such a system is in statistical control. You can make predictions for such a system with reasonable certainty (never 100%). Assuming you can make more accurate predictions by using more detail in your schedule plans demonstrates a lack of understanding variation and does not work. Do you know if your project delivery system is stable? You can use process control charts for variables such as on-time delivery to answer this question.

A system in statistical control usually shows only what is known as common-cause variation (i.e., random variation inherent in the system itself). Faults from fleeting events, with a definable cause that can be prevented or removed, are special-causes. Deming asserts common-cause variation accounts for 94% of variation. Only management can work on the process to reduce common-cause variation: they cannot blame the workers, and cannot assign "belts" to reduce it for them.

While all things have causes, responding to random variation as if it had a discrete cause leads to ever increasing error. Two mistakes must happen when managing variation:

1. To react to an outcome as if it came from a special cause, when actually it came from common-cause variation;
2. To react to an outcome as if it came from common-cause variation, when it actually came from a special cause.

Process behavior charts provide the tool to minimize both mistakes.

LPM/CCPM uses buffers to manage common-cause variation and project risk management to help control special causes.

One useful process behavior chart for a multiproject environment plots the final buffer penetration as projects complete. Figure 9.10 illustrates a chart of real data for a series of projects upgrading the control system on a nuclear reactor. You may use various kinds of process behavior charts. The one shown and perhaps the most common kind is called an XMR chart. X is the variable you measure which in this case is the buffer penetration at project completion. MR means that you use the moving range of the values to determine the process common-cause variation. As the figure shows, process behavior charts come in pairs: one chart for the moving range and one chart for the variable X. The inset illustrates how to create the chart from the data. You can set up an Excel spreadsheet to create the charts easily.

Various signals will tell you when your process needs attention. The most common and simplest is a single point outside the limits shown on the chart. Other signals include upwards or downwards trends of the data and other groupings of the data. I highly recommend reading Wheeler's work (2006) as you delve into this.

The buffer penetration process control chart tells you when an individual project likely had a special cause of variation. You should follow up on that project with a causal analysis and then remove the cause from the system so it does not affect future projects. The chart also tells you the total range of your current process. You can plan process improvements to reduce the variation in the project tasks that contribute the most to buffer penetration and thereby reduce the range. As you reduce the range of the variation you may be able to revise your buffer sizing policy.

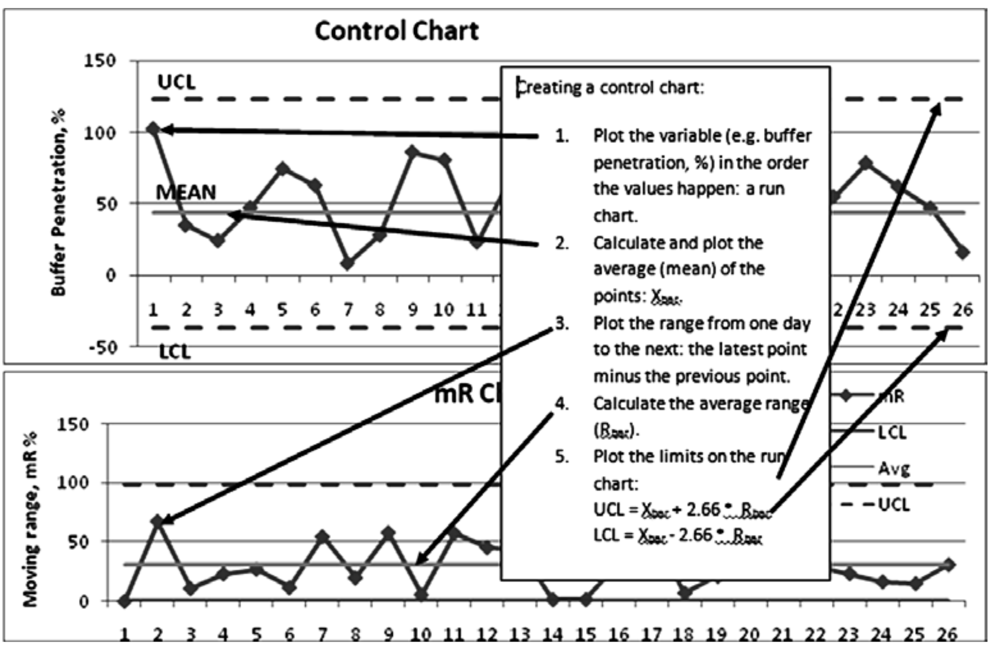


Figure 9.10 Process behavior charts enable appropriate system decisions. The process behavior chart limits show common-cause variation. Management must change the process to reduce that variation. Points outside the limits signal likely special causes, which the process operators (with management help) can work to reduce.

Analyze the process behavior charts to decide what action you need to take on your project control system, including setting new fever chart boundaries. The first steps are:

1. Determine if the process is stable (i.e., few points outside the control limits).
2. Determine if the range of variation is acceptable. It should be less than ±100% of the average from lower control limit (LCL) to upper control limit (UCL).

If the process is not stable, you must remove the causes of the out of control points. If the process is stable, but the variation too large, you must reduce variation: use the Six Sigma tools to do so. Start by confirming that task behavior expectations have been achieved (e.g., using buffer-based task dispatching, little or no multitasking, and no evidence of Parkinson's law). If you cannot take actions that achieve control within +100% of your current buffers, then you must revise your buffer sizing policy to increase the relative buffer.

You can use variations of the buffer process behavior chart to gain early warning. For example, you could create charts that track for each project the buffer penetration when the project has completed 25%, 50%, and 75% of the critical chain. That might help you determine where your projects are most variable. You might also create these charts for different groupings of projects. One former client of mine did this and found significant differences between project types on when in the project they used the most buffer. This helped focus improvement. The 25% chart might also give you early warning on a change affecting many projects that may not otherwise be detectable until it is too late to do much.

You can use process behavior charts for other measures: nearly anything you need to measure is subject to a form of the process behavior chart. For example, you might track total WIP (number of projects started but not finished) to monitor and control your Pipelining process. You can use them for many of the measures suggested above for each role. You might also use process behavior charts for the Kanban measures addressed next.

9.6 Kanban Measurements

A set of measurements has been evolving for local Kanban use. Anderson (2010) described several of them, including:

1. Trending of WIP by Kanban board and total for all Kanban boards.
2. Transit time for tasks on Kanban boards (also known as lead time for project and nonproject work tasks).
3. Due date performance for nonproject work.
4. Throughput of each Kanban board.
5. Issues and blocked work items identified on each Kanban board and total.
6. Flow efficiency. This is a measure of touch-time to total time for a task to transit a Kanban board. It is an indicator of multitasking and synchronization because queue time and blocked time are included in transit time.
7. Initial quality, usually measured by defects.

8. Failure load: how many Kanban tasks represent rework.

You need to select those which provide the results measures and behavior feedback your system needs while avoiding creating waste by the measurement system itself. Some of the electronic Kanban systems automatically track a variety of metrics from which you can choose for control.

Additional measures can help you to coordinate the local Kanban processes with the multiproject system:

1. *Backlog:* all the tasks in the pipeline for that Task Manager (or resource).
2. *Input queue:* project tasks with complete predecessors.
3. *Class:* red/yellow/green task buffer impact.
4. Nonproject work given class (red/yellow/green) when it emerges.

In each case before you deploy a metric, you should be clear on the question you are asking and what will signal a need to take action. For the latter, you may consider process behavior charts for each of the metrics.

9.7 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 6.6 presented 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.

9.8 Change Control Actions

Section 6.11 described 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 their Project Plan, including the schedule. You should have additional approvals for Project Plan changes that affect client deliverables (e.g., scheduled completion date, overall budget, or scope). 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 schedule logic (e.g., adding a task, deleting a task, making a change to the predecessor or successor on a task) should require formal change approval.
- 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-buffer, feeding-buffer, 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 execution to the plan, not on managing the plan. Do not, for example, make changes to your buffers based on actual performance to date.

9.9 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 reschedule 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's 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.

   If this happens on multiple projects, you may have an opportunity to improve your organization's planning process and execution of projects.

It may suggest your task duration estimates need to be reduced for future projects.

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 have not even started work on my task yet and it is 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.
6. *How do we prioritize nonproject work relative to project tasks?* Nonproject work seems to always come with a need-it-now priority. Your work process must establish business rules for prioritizing nonproject work relative to project work. You should track the amount of nonproject work in the system and, as suggested above, the priorities assigned to nonproject to bring them into the queues of Kanban boards also performing project work. Management should monitor these measures and adjust behaviors as necessary to support the organization's goal.

9.10 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. Consider both clients and project-team members as customers of your project-reporting-and-control system. Because buffer reporting must be timely to be effective, you should status tasks as they start and finish and at least once a week to ensure that accurate information is always available to all users for deciding which tasks to pull for execution. The most important items in this chapter include the following:

- 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 Managers use the information to decide when to take action on a project. Resource Managers use the information for longer-term resource decisions.
- Focused project meetings, using previously statused schedules, are a powerful tool to move your project to successful completion.
- Buffer-recovery planning and execution are essential parts of the CCPM control process.
- If cost is important to your project, you can use the buffer and earned value (EV) to derive cost-buffer penetration but do not attempt to use earned value for schedule control.
- CCPM puts a premium on project quality management and thus requires effective quality metrics.
- The Kanban processes require metrics for control and improvement.
- Conventional project change control methods are necessary to handle scope changes and the impacts of special-cause variation.

CCPM users find implementation of buffer management to be relatively simple and a very effective overall approach to project management and control.

References

CCPM+ Software. http://www.advanced-projects.com/CCPM+.htm. Accessed September 2013.

Daniels, A., and Daniels, J. (2006). *Performance Management*. Atlanta, GA: Performance Management Publications.

Goldratt, Eliyahu M. (1990). *The Haystack Syndrome*, Croton-on Hudson, NY: North River Press.

Herroelen, W., R. Leus, and E. Demeulemeester. (2002). "Critical Chain Project Scheduling: Do Not Oversimplify." *Project Management Journal*, Vol. 33, No. 4, pp. 48–60.

Ireland, Lewis R. (1991). *Quality Management for Projects and Programs*. Upper Darby, PA: PMI.

Juran, Joseph J. (1988). *Juran on Planning for Quality*, New York: The Free Press.

Ohno, T. (1988). *Toyota Production System*. Portland, OR: Productivity Press.

PMI. (2005). *Practice Standard for Earned Value*. Upper Darby, PA: Project Management Institute.

PMI. (2007). *Project Manager Competency Development Framework,* 2nd ed. Newtown Square, PA: Project Management Institute.

PMI. (2013). *A Guide to the Project Management Body of Knowledge,* 5th ed. Upper Darby, PA: Project Management Institute.

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

Womack, James. (2011). *Gemba Walks*. Cambridge, MA: Lean Enterprise Institute.

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 that reality in the future will follow your assumption. If your assumption does not come true, you have a project-risk event.

The PMBOK™ Guide (PMI, 2013) and *The Practice Standard for Risk Management* (PMI, 2009) suggest that risk management can include opportunities as well as negative consequences. They defined project risk as: "an uncertain event or condition that, if it occurs, has a positive or negative effect on a project's objectives." 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 mentally change that which follows if you are considering positive risks.

Project managers have several options to deal with project-risk events, including:

1. Applying 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 with CCPM risk management need only deal with special-cause risks. The CCPM process provides the necessary and sufficient process and tools to deal with common-cause risks potentially affecting schedule and cost and those potentially affecting scope to some degree. The project quality process is also a risk-management tool, protecting the project from scope risk.

PMI's updated *Practice Standard for Risk Management* (PMI, 2009) now discriminates between common-cause variation and special-cause variation when addressing project risk. I feel that this is a major step forward. Alas, its Appendix D provides some "strengths and weaknesses of CCPM," which suggest incomplete understanding of CCPM. In particular, it states, "Feeding buffers can deal with predictable common-cause variation, but may be inadequate to cater for special-causes." The intention is that buffers (project, feeding, capacity constraint time buffers, and the cost buffer) are all to deal only with common-cause variation. However, although not their intent, once you include buffers, they can also help deal with a certain amount of special-cause variation. Risk management as described in this chapter deals with special-cause variation.

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 that 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 (MORT). 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 only quantitatively useful if you can insure against the risk. Otherwise, it is going to cost you nothing or $100,000, but 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 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.

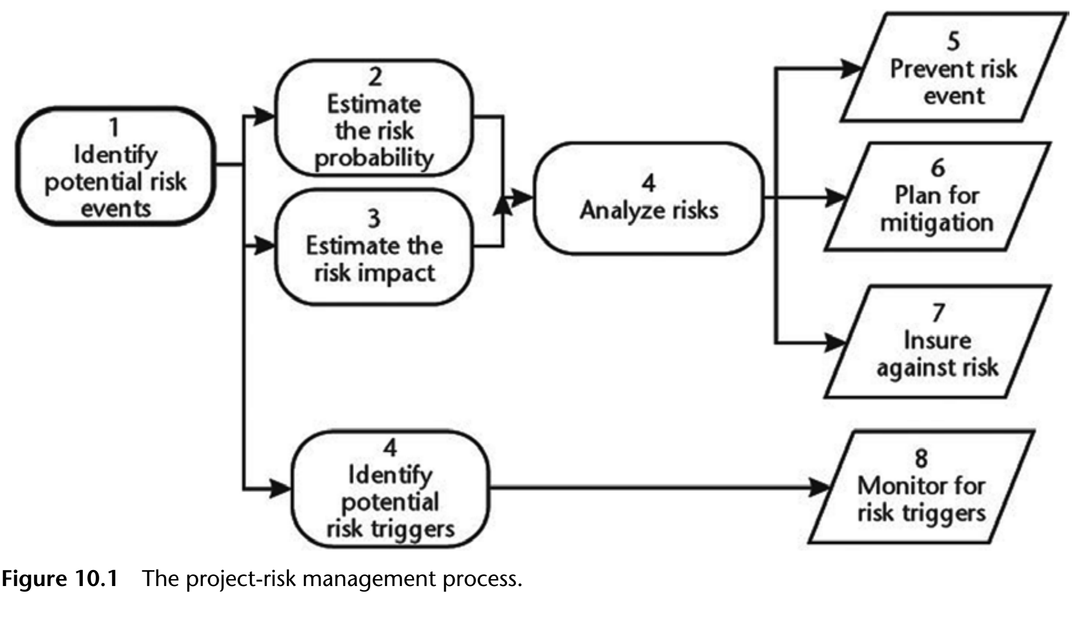


Figure 10.1 The project-risk management process.

Relatively small projects (i.e., less than a few million dollars and under one year) should not have a risk list in excess of 10 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 first 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 consequence, 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 (2004) provided 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>." The example risk statements in Table 10.1 use boldface for the cause and consequence and italics for the effect to illustrate how to use the format.

Columns 2, 3, and 4 provide a relative quantification of risk. Risk is the product of consequence and probability. The table footnote provides one possible 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 fifth 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 indicators whenever possible.

Columns 6 and 7 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. However, 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 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:

Table 10.1 The Risk Matrix

Risk Event	*P*	*C*	*R*	*Trigger to Monitor*	*Prevention Actions*	*Mitigation Actions*
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
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 noncombustbles Fireproof storage cabinets	Install fire suppression ASAP
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
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
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

P (Probability): 3 > 20 to <50%; 2 = 5–20%; 1 < 5%;
C (Consequence): 3 > project buffer; 2 ≥ 20% project buffer; 1 < 20% project buffer.
R = P * C.

- Preventing (or reduce the likelihood of) the risk event (e.g., breaking the project into phases or researching uncertain project elements to improve certainty);
- Scheduling high-risk activities early in the project;
- 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 those illustrated in 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 that your team felt necessary 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 (Wideman, 1992). You may evaluate the problems encountered by previous similar projects. You may simply get your project team together and brainstorm a list to start with (this is the approach I usually take). 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.

Heath and Heath (Heath, 2013) provided some useful insights and suggestions based on psychological research that can help your risk identification and management. Although their work focuses on decision-making, the psychological facts on which they build and the recommendations that they make apply to several aspects of risk management. For risk identification, they identify the rule proposed by Kahneman (2011) that "what you see is all there is." In other words, it takes mental work to go beyond the bounds of the information presented to you to consider potential risks to your projects. Tools that they suggest include "bookending the future," Failure Modes and Effects Analysis (FMEA), and project "premortems." Premortems mentally place yourself at the end of the project and consider that it has failed (or succeeded) and you need to investigate why. Other psychological cognitive biases include seeking confirmatory evidence instead of looking for things that will prove you wrong and being overconfident in estimates. If you do not think that you have these biases, you are suffering from the latter. I strongly recommend reading *Decisive*.

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.

However, 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 Project 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 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 1 or 2 years in duration), the list should be less than 10 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 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. However, 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 (1996) noted, "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 is observed, that is, where the laws of chance work in favor of the insurance company or gambling institution. The very nature of risk then ensures that we are dealing with relatively low-probability events to start with.

People's ability to estimate probability is notoriously poor (Kahneman, 1982; Kahneman, 2011; Belsky, 1999; Russo, 1989). When considering probability, most people are subject to numerous logical biases and errors. Unfortunately, research demonstrates that 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. Because 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 that people associate with schoolteachers. Because 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 he or she is given to review (i.e., he or she is 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 are 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 more than a moderate risk, which you might define as ranging from a 5% to 20% 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 estimate the risk consequence to define the risk because risk is the product of probability times consequence. You could qualify consequence in terms of the overall project schedule and cost or the expected return on investment for the project. CCPM provides a unique measure of estimating the risk consequence 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 Consequences (3)

A high-impact consequence 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 Consequences (2)

A moderate-impact consequence is an impact that would consume on the order of 20% of your project buffers.

10.3.3.3 Low-Impact Consequences (1)

Estimated consequences would not exceed about 20% 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 to monitor for the risks that 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 imminent, if risks are past, or if new risks are perceived. Sometimes you may need more formal monitoring for the risk triggers.

Heath and Heath (2012) discussed monitoring in their Chapter 11. They noted (p. 226), "Because day-to-day change is gradual, even imperceptible, it's hard to know when to jump. Tripwires tell you when to jump." Of course, the buffer action criteria provide one new set of tripwires within CCPM. You should look for additional ones for specific risks.

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 Summary

Project risk management controls the impact of 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 for risk management 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.
- 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

Belsky, Gary, and Thomas Gilovich. (1999). *Why Smart People Make Big Money Mistakes, and How to Correct Them*. New York: Simon & Schuster.

Bernstein, Peter L. (1996). *Against the Gods: The Remarkable Story of Risk*. New York: John Wiley & Sons.

Heath, C., and D. Heath. (2013). *Decisive*. Haryana, India: Replika Press.

Hilson, David. (2004). *When Is a Risk Not a Risk: Part 2*. http://www.riskdoctor.com/pdf-briefings/risk-doctor07e.pdf. Accessed June 22, 2004.

Kahneman, Daniel, Paul Slovic, and Amos Tversky. (1982). *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge, U.K.: Cambridge University Press.

Kahneman, Daniel. (2011). *Thinking, Fast and Slow*. Cambridge, U.K.: Cambridge University Press.

PMI. (2009). *Practice Standard for Project Risk Management*. Newtown Square, PA: PMI.

PMI. (2013). *A Guide to the Project Management Body of Knowledge,* 5th ed. Newtown Square, PA: PMI.

Russo, J. Edward, and Paul J. H. Schoemaker. (1989). *Decision Traps: The Ten Barriers to Brilliant Decision Making, and How to Overcome Them*. New York: Simon & Schuster.

Wideman, R. Max. (1992). *Project and Program Risk Management: A Guide to Managing Project Risk and Opportunities*. Newtown Square, PA: PMI.

Implementing the Change to CCPM

Most organizations with which I deal feel that they are pretty good at project management. They have been doing it a long time. Most people with the job title of project manager feel they are pretty good at it. My exposure to dozens of organizations suggests otherwise to both. While there are indeed some organizations that excel at project management, and there are some excellent project managers, my exposure to a wide variety of business and government organizations suggests that they are in the minority.

A well-researched basis for this known as the optimism bias applies (Sharot, 2011). A large majority of people perceive themselves as being superior to others (i.e., better than average) on just about anything whereas, mathematically, only 50% can be better than average. Sharot reported studies showing 85% of people rank themselves above the 50th percentile for the ability to get along with others and 93% of people believed they were in the top 50% of driving ability (p. 15). Similar statistics seem to apply to people's estimates of their organization's project management capability and performance and their own performance as project managers.

While it is easier to succeed at CCPM than at some forms of conventional project management, it requires an organization that sustains necessary project management basic practices such as:

- Project Charters;
- Project Plans;
- Project change control;
- Management to effective resource-loaded and leveled dynamic project schedules.

If your organization does not use practices that ensure effective application of such processes, you might consider yourself as leading the implementation of real project management in the organization rather than just a change to a different way of execution.

I have yet to come across an organization that applies constraint-driven demand-leveling, also known as WIP control, to their overall project system. I have not come across an organization where management shares a consensus on what the constraint is to project throughput. Recall that the first TOC focusing step is to identify. The second and third focusing steps are to exploit and subordinate (i.e., first work to get more effective results from what you have). Unfortunately, if pressed, most managers will immediately jump to TOC focusing step 4, Elevate, that is, we need more X, rather than the solution of applying the exploit and subordinate directions offered by TOC and CCPM. For clarification on this, please refer back to the five focusing steps addressed in Section 3.9.3. CCPM does not require additional resources and need not take much time.

The time required for organizational change to produce desired results depends 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 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.

Most larger organizations take a year or longer to get all of the projects planned as CCPM projects and make the necessary management behavior changes. That may seem too long to some but is short in terms of the above examples. The following presents a process that, with diligent leadership, works for both single-project and multiproject organizations. I then present the theory underlying the approach.

11.1 Rule 1: Focus with Kanban

I used to recommend that organizations start with an implementation model for enhanced project delivery as described in the next section. In recent years with my discovery of project Kanban, I now recommend Kanban as the place to start. It enables you to work on reducing the waste of multitasking and reduce actual WIP immediately. It is much easier to do than to change an organization to apply professional project management tools and usually gains immediate acceptance within organizations. It costs virtually nothing if you apply the common sticky note approach. You can start today in any workgroup without tools and in most cases without anyone's approval. It is flexible enough to accommodate any kind of work and need not be applied consistently across the organization to yield local results from behaviors that are consistent with the desired global result (i.e., Rule 1: focus and WIP reduction).

While workgroup level Kanban will not cause much in the way of bottom-line impacts for the organization, it begins the process of increasing workflow through controlling local WIP[1] and applying the principles of pull and visual control. Kanban boards provide a visual means to see all of the work, both project and nonproject, that workgroups are handling and puts the project and nonproject work on an equal basis for entering the workgroup's active queue. Although the project system effectiveness will improve when there are effective resource-leveled and synchronized (Rule 3: Pipelined) project schedules for task input, workgroup Kanban sets the basis for using the task priorities that will flow from that system.

Anderson (2011) suggested five core principles for successful Kanban applications:

1. Visualize the workflow and the work.
2. Limit WIP.
3. Manage workflow.
4. Make process policies explicit.
5. Improve collaboratively (using models and scientific methods).

I have explained the basic project Kanban processes earlier in this book. If you are in software development or are able to ignore the software development context of some of the discussion, I highly recommend David Anderson's book (Anderson, 2010) and his Web site as powerful resources to get your teams started; 99% of his thinking applies to any type of work. You can then apply the principles and approaches I describe to your leading Kanban implementation.

11.2 Rule 2: Buffer with CCPM

Figure 11.1 illustrates a basic project model to implement CCPM. Implementing CCPM in an organization is a project. Implementation plans vary in content and scope due to the specific organization. The rest of this section describes an example implementation plan. Following the example plan, Section 11.4 describes some of the theory underlying it.

Figure 11.2 illustrates a more thorough road map of the content you might include in your workplan. Please do not let it intimidate you. You do not have to

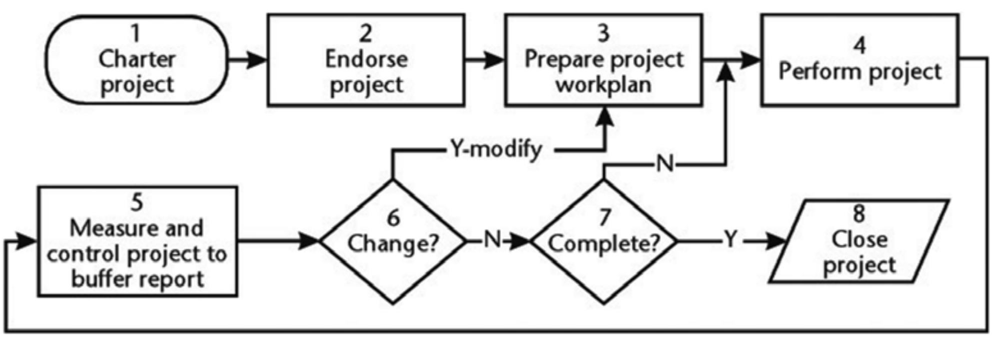


Figure 11.1 Implementation process flowchart.

1. If management has not reduced project WIP by suspending some started projects, Kanban does nothing to reduce organization WIP. It just identifies more stuff in backlogs than on Kanban boards.

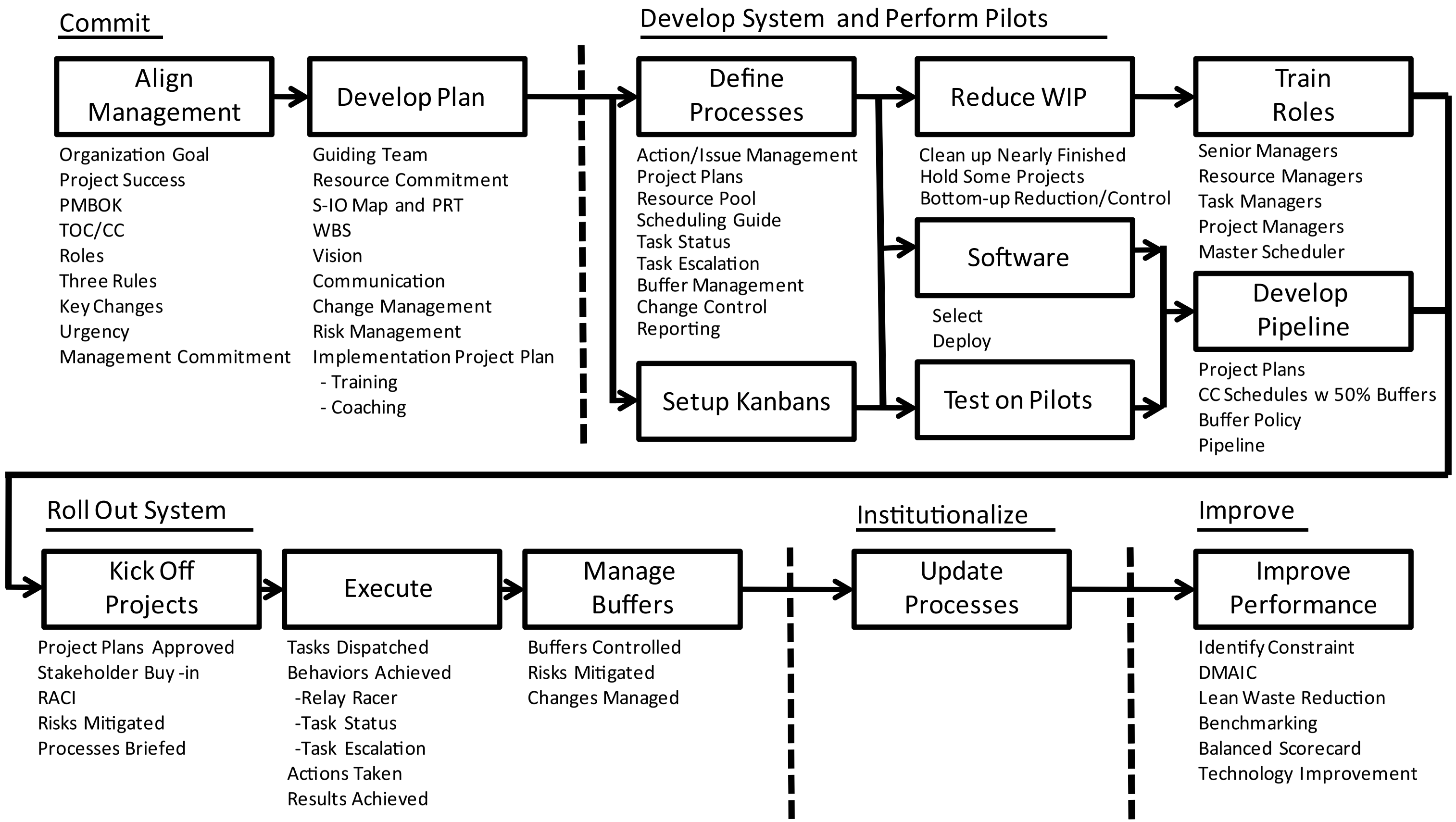


Figure 11.2 Implementation road map.

do all of it. It is here to trigger your thinking so you can decide on the things you need to focus on in your plan.

I recommend you read through the overall flow boxes of Figure 11.2 first and then go back and examine the detail items listed under each box. The blocks are separated into five phases:

1. Commit.
2. Develop system and perform pilots.
3. Roll out system.
4. Institutionalize.
5. Improve.

The following discussion addresses most of the detail items and the others are addressed elsewhere in this text. Please use the index to help you find them.

11.2.1 Endorse the Implementation Project

The Commit section of Figure 11.2 addresses J. Kotter's first two (of eight) points of successful change programs: establish a sense of urgency and create a powerful guiding coalition (Kotter, 2002). 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. As 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.

One challenge CCPM faces with senior management endorsement in many organizations is that unlike Six Sigma and Lean, CCPM does not promise cost savings. This has been a challenge for other TOC-based improvements as well. Most senior managers feel extreme pressure to cut costs. The TOC way is to make the company much more profitable by expanding throughput. Most functional managers in large organizations do not feel that increasing throughput is in their control. However, they all can cut costs in their function, sometimes even at the expense of organization throughput. Here is an illustration I learned from Bob Fox and that he presented in *Profitability with No Boundaries* (Pirasteh and Fox, 2011, pp. 169–171)[2] that might help you convince them of the hole they are digging with cost-world thinking.

Suppose the second column of Table 11.1 represents your organization's present profitability using TOC accounting. Assume you come up with a way to increase direct labor productivity by 20%. You decide to exploit that huge potential cost savings by laying off 20% of your labor force. Direct labor cost goes down by

2. I have changed the model slightly from that presented in the book to keep the direct labor the same for case 2.

Table 11.1 Increasing Throughput Is Much More Effective Than Decreasing Cost

	Initial Case	*Reduce Direct Labor*	*Increase Throughput*
Sales	$100	$100	$120
Raw materials	$35	$35	$42
Direct labor	$10	$8	$10
Overhead	$50	$50	$50
Total cost	$95	$93	$102
Net profit	$5	$7	$18
Increase		40%	260%

20%. Sales and other costs remain the same. You are able to report a 40% increase in profits. Isn't that great?

However, let's suppose instead your management decided to exploit the productivity increase by increasing throughput 20%. In that case, you do not realize the direct labor cost savings and your raw materials cost actually increases by 20%, but net profit increases to more than three times the base case, a 260% improvement. Think of what that would do to the stock price and senior management compensation.

I understand that even the most senior level functional managers will argue that they have the ability to reduce cost in their department but that only marketing can increase throughput and that marketing will argue that they cannot increase sales because they are a limited by the market. You can see the huge problem that such local performance thinking causes in organizations; they forgo a factor-of-3.6 increase in profit because of the way they have chosen to organize and focus. Also note that even if they did not have the ability to increase direct labor productivity, the increase in net profits from a 20% increase in throughput still vastly outweighs a 20% cost savings. I do not pretend to have a chance to cause that level of revolution in management thinking, but perhaps this illustration will help you cause some of them to think of possibilities.

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.

One of my former clients who guided one of the more successful sustained CCPM implementations that has delivered huge results to his company noted to me, "After almost four years of managing projects by critical chain principles, I would say most of our task managers still see their CCPM responsibilities as an add-on to their jobs rather than a complete transformation of their jobs." That means that their managers have not realized the need to create new katas in the company. That is why senior management must endorse the right principles.

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. However, I have also noted that in today's organizations, active support by the formal organization leader is rarely enough. Kotter (1998) noted (p. 6):

> 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.

Kotter noted in his most famous work, *The Heart of Change* (Kotter, 2002, p. 2):

> The central challenge…is *changing people's behavior*. The central challenge is not strategy, not systems, not culture. These elements and many others can be very important, but the core problem without question is behavior—what people do…Changing behavior is less a matter of giving people analysis to influence their thoughts than helping them to *see a truth to influence their feelings*…the heart of change is emotions.

Kotter builds on the latter idea by organizing his eight points in terms of see→feel→change instead of analysis→think→change. You might note that this is a radical departure from Dr. Goldratt's analytical approach and might help to explain why Goldratt's methods have been less successful in larger organizations.

My experience with implementing CCPM fully supports Kotter's assertions.

11.2.2 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.

11.2.3 Begin with the End in Mind (Vision)

If you do not know where you are going, you will not 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 because many people respond better to 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 projects successfully, because it is all about changing behavior. Some organizations make the mistake of thinking CCPM is about buying and implementing software. It is not. You need to create a mental picture of running and winning the relay race or the flow of work like a river as compared to people buried in WIP.

....

Project Charter
Project: Implement Critical Chain Project Management
Revision: 0 Date: 1/1/2015
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 to enable roadrunner task performance that achieves a project Throughput increase of 100% and cycle time reduction of 50% with reduced quality defects of more than 30% accompanied by an initial Work in Progress (WIP) reduction of at least 50%.

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:
WIP reduction and initial use of CCPM on pilot projects is expected to be substantially complete within 90 days of the approval of this project Charter. Quarterly progress reviews are to be held for the following three quarters (i.e., the final one is to be held on January 1, 2015).

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 rescheduling of projects using CCPM and buffer management are not included in this cost as they are part of the respective projects and minimal where adequate project schedules exist.

Special considerations:
Procedures and software tools should comply with company format and computing capability.

Acceptance: ____________________________, *Project Manager*

....

Kotter's fourth and fifth steps 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 act on the vision, including overcoming the obstacles to change. Tables 11.2 through 11.6 list the various behavior changes to be expected from different groups.

11.2.4 Create the Implementation Project Plan

The Project 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 11.2 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 and without resourced schedules	Only commit to delivery dates with a project plan using Critical Chain schedules 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	Prioritize all work for execution using the Kanban processes
	Prioritize demands for routine administrative work above project work (e.g., salary reviews)	Assure sufficient capacity constraint buffer for administrative work and time off
Set project priority*	Lack clear project priority or have changing project priorities	Set project priorities for Pipelining including the priority of new projects relative to ongoing projects
Control project Work in Progress (WIP)	Do not control the project WIP to the capacity of the organization constraint	Reduce and hold WIP low to increase project throughput and reduce cycle time
Select drum resource* for pipelining	Fail to consider system constraint	Select the drum resource to be used for sequencing the start of projects and creating the drum schedule
Select master scheduler and approve project sequencing*(pipelining)	Start each project independently as funding is available, or start all projects at the beginning of the year	Senior management sets project priority, approves Pipeline including WIP limits
Review project status	Listen to project status reports and seek blame for problems.	Proactively aid buffer recovery

*Only for multiple projects.

Table 11.3 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 or, better yet, enable resources to pull tasks using buffer information
Do resource planning	Plan resources by name and task	Plan resources by type and workgroup (task manager) and enable resources to pull tasks for execution using buffer information
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 Control WIP at the workgroup level
Generate resource buffers	Schedule resources far ahead that are then not available when needed	Use task priority list to enable resources to pull tasks for execution based on buffer penetration.

Table 11.4 Project Managers' Behavior Changes

Change	*Present Behavior*	*Future Behavior*
Use 50% probable task-duration estimates	Project managers send a message that they expect due dates to be met	Project managers estimate task focused average durations and low risk durations, using the 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 schedule changes	Varies: often submit changes to minimize minor variances	Allow schedule changes only when triggered by buffer report or approved scope change
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-cause 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

*Only for multiple projects.

Table 11.5 Subcontractors' Behavior Changes

Change	*Present Behavior*	*Future Behavior*
Deliver to lead-times	Deliver to due dates	Deliver to lead times (i.e., only to buffered dates established with CCPM schedules)
Shorten lead times	Deliver to due dates	Shorten lead times

Table 11.6 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 precede all dates with a buffer

4. Resource-loaded (Critical Chain) project schedule;
5. Project budget;
6. Definition of the project team by lead person;
7. Procedures for operation of the project team;
8. Plans for project close out.

Figure 11.3 illustrates a WBS for a project to implement CCPM. The WBS reflects the changes necessary for CCPM and includes the overall approach recommended by Kotter (see Section 11.4.1). 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. Senior Management sets WIP limits and prioritizes projects and nonproject work.
3. The Master Scheduler creates the drum schedule to accommodate management's project priority.
4. Project Managers schedule projects to the drum schedule.
5. Resources work to the relay-racer paradigm.
6. Resources or Task Managers provide accurate input on the RDU of their tasks.

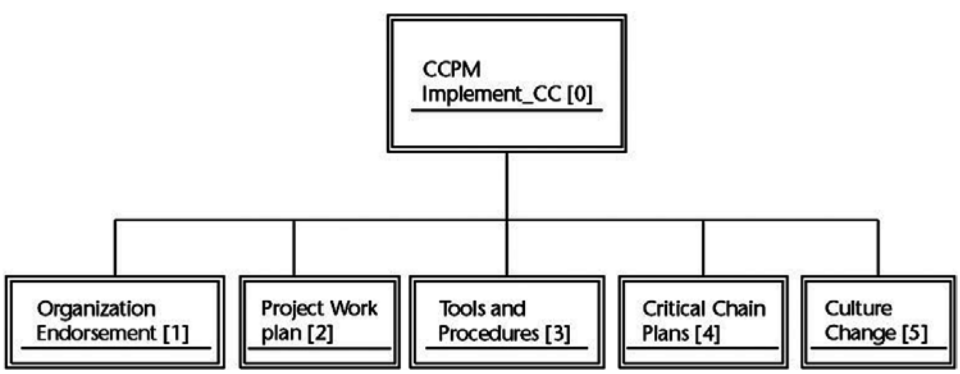


Figure 11.3 The WBS to implement CCPM identifies the work package deliverables.

7. Resources and Task Managers use the prioritized task lists to avoid bad multitasking and to determine which task to work on next.
8. Task Managers lead work execution to minimize buffer use.
9. 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 project schedule tasks developed following the WBS must create these results. 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 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."

You can create a schedule for implementing CCPM. If you do, it should be a Critical Chain schedule. Your implementation team can practice with statusing and controlling to the implementation schedule. If you also use Kanban for the implementation team, you can also pilot the interface between Kanban and a CCPM schedule.

The pilot projects dominate the overall implementation project duration and thus present a problem for scheduling the implementation. Pilot projects usually need not complete to serve their 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. Generally, this is after a fixed duration (e.g., three months). Because you have to do preliminary work to get ready for the pilot projects and they feed into the pilot project run duration, your critical chain would include this long fixed duration task. That is acceptable as long as you do not include that fixed duration when computing the project buffer for the implementation project.

Alternatively, you could plan your implementation as a mini-pipeline of a project to get ready for the pilots, the fixed duration pilot time, and a follow-on project to implement the learning from the pilot projects and roll out the resulting system organization-wide.

If you have a large organization, the number of projects to be planned may be quite large. Be sure to perform WIP reduction before deciding which projects to plan in detail for the Pipeline. 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 Critical Chain compatible project scheduling software can take very long, but you should work to keep that process off the critical chain of your implementation if possible. Consider focusing on behaviors and doing the pilot projects with your existing critical path software.

11.2.5 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 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 are 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 11.7 presents an example CCPM risk-management matrix, using the format described in Chapter 10 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 third through fifth 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.

11.2.6 Just Do It or Fake It Until You Make It

Tables 11.8 through 11.10 lay out the steps necessary to implement CCPM in a large, multiproject environment. Scale these steps to fit your needs. The most

Table 11.7 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 the pilot project to ensure that 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 to train and coach.
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.

P (Probability): 3 = 20–50%, 2 = 5–20%, and 1 5%.

C (Consequence): 3 is > the project buffer, 2 is 20%> the project buffer, and 1 is <20% and ≥ the project buffer.

R = P * C.

important first step is to select the right people for the leadership team. Kotter's definition of a "powerful guiding coalition" applies.

Your team can accomplish all of the actions in Table 11.8 in a one-day meeting. This meeting best follows a two-day workshop where all of the managers learn the Critical Chain theory. It is a mistake to separate the two-day workshop from this implementation meeting, as people forget training very rapidly if it is not immediately reinforced by application in the field.

Table 11.9 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 planning sessions should not take more than a few weeks in total. If it is taking longer than that, it is likely your teams lack the necessary skills to effectively plan their projects and you need to focus on coaching them on how to do it. If there are many projects, you may need to have multiple facilitators so that this step does not drag out.

Table 11.10 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. You need to use the coaching kata to instill the coaching kata into the managers

Table 11.8 Implement Phase 1

Step	*Responsibility*	*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	
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 including an overall WIP reduction to start	Senior management commitment to implementation Senior management agreement to WIP limit for pipelining and initial projects to place on hold
10	Leaders	Determine project priority (or sequence for the drum resource)	Senior management agreement to 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	Master scheduler	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

Table 11.9 Implement Phase 2: Individual Project Critical Chain Plans

Step	*Responsibility*	*Action*	*Output*
1	Facilitator	Deliver 2-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 chain 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)

that are going to make project delivery to the CCPM paradigm the normal culture of the business.

Table 11.10 Implement Phase 3: Drum Schedule and Project Schedules

Step	*Responsibility*	*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

11.2.7 Measure-and-Control Implementation

Measurement 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 Kanban meetings and weekly buffer meeting are the primary vehicles for this feedback. They are 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. Pipeline sequencing of project starts eliminates much of the serious resource contention for the project teams.
3. Buffer management provides a clear decision-making tool to allocate resources among 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.

Management can enhance the effect of these natural feedbacks (positive behavior reinforcement) 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 reinforcement for increased project success.
4. Management experiences positive feedback for increased profitability.

Project Kanban can begin the process with item 1 above. While some of the other 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 (see Sections 9.2 and 9.3);
2. Ensure that any new projects posed for inclusion into the system are prioritized and fit into the drum schedule.

11.2.8 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 Project 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 CCPM," 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.

You are stuck if you have been working on the implementation for more than three months and are not essentially there and beginning to get positive feedback. Kotter suggests you need to go back and work on the sense of urgency. You need to dig into your organization's policies, measurements, and behavior to find out where and why you are stuck and implement remedies to remove the block(s). From a TOC perspective it means that management has not learned to subordinate their behavior to the organization's throughput constraint. 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 (confirmed for me by my own observations) I suggest that you start by looking at your leadership.

11.3 Rule 3: Pipeline to Maintain Low WIP

Although I address Pipelining above as part of an overall plan using individual CCPM schedules for Pipelining, management usually needs to take some preemptive steps to begin the process of demand-leveling for the organization. Most CCPM practitioners agree that all of the organizations they work with have too much project WIP. Management action to summarily reduce project WIP at the outset enables taking much waste out of the system and thereby increasing project flow and throughput. It demonstrates a serious engagement by Senior Management to change the way the system operates. Most CCPM practitioners recommend putting anywhere from 25% to 50% of the active projects on hold and not restarting them until projects complete.

Alas, most management teams with which I have worked lack the fortitude to take such action at the outset. They rightfully point out that their customers will punish them for stopping their projects. Project customers (internal or external) will not believe the assertion that management is taking the necessary action to complete all projects sooner. Even though they can agree logically with everything that has been presented here and can see dozens of case studies of organizations that have succeeded with this approach, they will tell you all the reasons that they are different. While their organizations certainly are unique, they are not different in this regard. Reducing WIP reduces waste, increases project flow, reduces project duration, reduces quality errors, and increases project throughput so that all projects finish sooner by suspending work on a bunch of them and delaying restart or new starts until projects finish. Reducing WIP always works if your organization is presently operating on the steep part of the queuing curve. Almost all organizations start there. I have never seen it work if you do not reduce WIP.

Even after you have demonstrated the effectiveness of CCPM and Pipelining, there will be constant pressure to increase WIP. If you allow it to happen, your projects will begin to run into the red buffer regions more and more and the waste of multitasking will reappear. Count on it.

11.4 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 is not 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 (Dalzeil and Schoonover, 1988) noted, "Technocratic leaders … focus exclusively on outcomes without considering the concerns of employees who must implement and sustain change." Many project managers are technocratic leaders (myself included) are subject to this blind spot. Dalziel and Schoonover further noted, "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.

I have come to rely on three major sources for knowledge on planning and executing CCPM implementation. I rely on them because they base their work on extensive research and field applications and because their ideas work. The sources are John Kotter, the Prosci organization's publications, and the Heath brothers' work. The following gives an overview of each of their ideas, but I strongly encourage you to dig into their reference works.

The Project Management Institute published their own guide to change management as I was completing this book: *Managing Change in Organizations: A Practice Guide* (PMI, 2013). It ties much change management experience with the PMIs project, program, and portfolio models and provides sources that go beyond those I mention. You may find it useful in addition to the three sources I describe in more detail below.

11.4.1 Kotter's Model

Kotter started with a survey of over 200 people in more than 90 organizations around the world that have tried to produce productive organization change. Kotter and Cohen followed up on preliminary results to create a testable model of how to produce effective change. He stated, "In The Heart of Change we dig into the core problem people face [when making organization changes]…changing the behavior of people." He made a strong case that the technocratic approach to change of Analysis→Think→Change almost never works. What does work is See→Feel→Change. Visualizations, including dynamic graphs and other visual controls, create feelings that reinforce new desired behaviors.

Figure 11.4 illustrates the eight-step process that Kotter and Cohen (Cohen, 2005) developed. Although drawn as a linear process without loops, it usually works with many loops and repetition of steps.

Here is a little more detail on each of the eight steps:

1. *Create urgency:* The first step is to get a sense of urgency building within the organization that something must be done. Most people in most organizations feel overwhelmed by keeping up with the daily workload so suggesting doing something more to change the way the organization works looks at best to be just more work and at worst to be something that is going to make things worse than the presently are. People need something to motivate them. Alas, not too many people in an organization are motivated by making more money for the stockholders or owners of the company, so you are not likely to build a sense of urgency below the second level of a company unless you go beyond bottom line impacts. Kotter (2002, p. 36) suggested some things that work:
   a. Show others the need for change with a compelling object that they can actually see, touch, and feel.
   b. Show people valid and dramatic evidence from outside the organization that demonstrates that change is required.
   c. Look constantly for cheap and easy ways to reduce complacency.
   d. Do not underestimate how much complacency, fear, and anger exists in your organization.

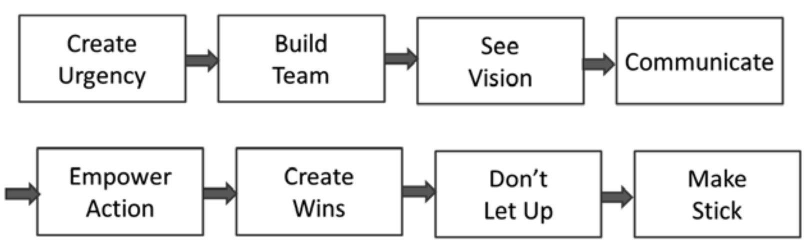


Figure 11.4 Kotter and Cohen's eight-step model for creating productive organizational change.

2. *Build team:* One person can only succeed to cause change in a very small organization. Many people are not even able to cause changed behavior in one person: themselves. Think of how many people succeed at losing weight or stopping smoking. So if you plan to create real change on an organization level, you are going to need help. You need to enlist the leaders of the organization who have bought into the sense of urgency. Kotter suggested (p. 60) the following list of some things that work:
   a. Show enthusiasm and commitment to help draw the right people into the group.
   b. Model the trust and teamwork needed in the group.
   c. Structure meetings for the guiding team to minimize frustration and increase trust.
   d. Put your energy into step 1 (raising energy) if you feel you cannot move on to step 2.
3. *See vision:* People need to be able to see the proposed change because that is what can begin to create an emotional feeling that will motivate them to change. A vision should be a picture of the end result. If you describe it in words, the words need to evoke an image. Kotter suggested the following things that work (p. 82, slightly revised):
   a. Try to see—literally—possible futures.
   b. Make the vision so clear that you can articulate in one minute or write, or better yet draw, it on one page.
   c. Supply a moving (emotional) vision such as serving people.
   d. Put forth bold strategies to make the vision real.
   e. Focus on how to quickly make the change.
4. *Communicate:* So people can feel the change, you need to communicate:
   a. The vision in terms of the benefits people will see when they change their behavior.
   b. What has to be done to make the vision a reality.
   c. Reinforcements when people exhibit the right new behaviors.
   d. "Wins" by people and groups who do the new behaviors.
   e. Anything and everything else about the change that will keep at the top of people's agenda.

   Kotter suggests the following ideas to help communicate (p. 101):
   a. Keep communication simple and heartfelt.
   b. Do your homework before communicating, especially to understand what people are feeling.
   c. Speak to anxieties, confusion, anger, and distrust.
   d. Rid the communication channels of junk so that important messages get through above the noise.
   e. Use current technologies to help people see the vision (electronic opportunities of all kinds).

5. *Empower action:* You need to empower action: make sure people know that they are expected to take action now and that they are free to do it as they see fit. Empowering action is as much about removing obstacles to action (pulling) as it is about causing people to act. Kotter suggested that the following things work (p. 122):
   a. Find individuals with change experience to bolster people's self-confidence with "we-won-you-can-too" stories.
   b. Recognize and reward in ways that inspire, promote optimism, and build self-confidence.
   c. Deal with disempowering managers through coaching or move them out of the way.
6. *Create wins:* Your team needs to coach people to create successes: wins. Then you need to reinforce the behavior of those who created the wins and communicate their wins and reinforcements to the rest of the organization. Pilots are a powerful tool to create short terms wins but you need to ensure that people who live those wins with the pilots do not immediately go back to prior behaviors. Kotter suggested the following (p. 140):
   a. Early wins that come fast.
   b. Wins that are as visible as possible to as many people as possible.
   c. Wins that go through emotional defenses.
   d. Wins that are meaningful.
   e. Early wins that speak to powerful players whom you need to engage.
   f. Wins that are cheap and easy even if small.
7. *Do not let up:* The leadership team has to keep the desired change at the top of agenda through and well beyond the planned-for successes. There will be obstacles and there will be some failures along the way but the winning teams take failure as a learning and motivating experience to add vigor to the change process. Kotter suggested that the following ideas help here (p. 159):
   a. Rid yourself of work that wears you down—tasks that mattered in past but may not matter now or tasks that you can delegate.
   b. Constantly look for ways to keep up the urgency.
   c. Use new situations opportunistically to launch the next waves of change.
   d. Show 'em, Show 'em, Show 'em…
8. *Make it stick:* Once you have completed the first round of getting the organization to exhibit the desired new behaviors, you need to continue right on to improve what you have accomplished. If you do not continue to improve, the organization will revert to the previous behaviors in a surprisingly short period of time. Kotter's ideas that work here include (p. 177):
   a. Never, never, never give up on step 7.
   b. Use new employee orientation to demonstrate what matters most in the organization.
   c. Use the promotion process to place people who exhibit the new behaviors into influential positions.

d. Tell vivid stories over and over about how things now work.

e. Ensure continuity of behavior and results that help sustain and grow the new culture.

And one he did not include: build on the improvement and coaching katas to make more improvements in the direction you have set.

Now go buy Kotter and Cohen's books and learn how to apply the details of what I summarized above. Cohen's book supplies useful specific steps you can take and provides a set of survey instruments to help you monitor how you are doing and make corrections.

11.4.2 Prosci's® Model

Prosci's Web site (prosci.com) gives the basis for their change management model, "The most recent study conducted in 2011 benefitted from 650 participants across six continents. Collectively, Prosci's proprietary research database and intellectual property includes studies completed on more than 2,600 organizations from 65 countries, including many of the largest companies worldwide. Prosci's access to participants is facilitated through the Change Management Learning Center, a Prosci-sponsored Web site designed to facilitate Change Management communication that has attracted approximately 65,000 registrants." You will find many useful articles on their learning center Web site: http://www.change-management.com/tutorials.htm.

Prosci builds on their ADKAR® model (Hiatt, 2006):

- Awareness of the need for change;
- Desire to participate and support the change;
- Knowledge on how to change;
- Ability to implement required skills and behaviors;
- Reinforcement to sustain the change.

Prosci breaks their process into three areas: preparing for change, managing change, and reinforcing change. Their research identified the following top six contributions to successful change projects (Hiatt and Creasey, 2012, Appendix B):

1. Active and visible executive sponsorship;
2. Frequent and open communication about the change;
3. Structured change management approach;
4. Dedicated change management resources and funding;
5. Employee engagement and participation;
6. Support from middle management.

The also identify the primary obstacle to successful change management came from senior leaders, including:

1. Inactive or invisible sponsors;
2. Poor alignment between organizational direction and the objectives of change;

3. Lack of sponsor commitment to change management;
4. Sponsors with competing priorities or changes in sponsorship;
5. Sponsors at the wrong level (not high enough) in the organization;
6. Little or no access to the primary sponsor;
7. Failure to build a coalition of sponsors.

They identify the top five greatest change obstacles as:

1. Ineffective change management sponsorship from senior leaders;
2. Insufficient change management resourcing;
3. Resistance to change from employees;
4. Middle management resistance;
5. Poor communications.

They offer solutions to all of these obstacles.

I find the research impressive and the model useful and supportive of Kotter's eight-step approach. Even though the R in ADKAR is for reinforcement, I do not see quite the emphasis on emotion and behavior reinforcement as Kotter, nor with the behavioral emphasis of the Heaths or Daniels (2006), but you might find specific help for your situation in Prosci's substantial learning center.

11.4.3 Heath Brothers' Model

The first book I read by the Heaths was *Made to Stick* (Heath, 2008). I was enthralled by it at the time because one issue I had been observing with CCPM implementations was that a discomforting fraction did not seem to stick. There seemed to me to be a high correlation between making ideas stick and making organizational change last. I observed that a pleasing fraction of organizations achieved some remarkable success in the first year with CCPM and some improved on that success in the second year but then a disappointing fraction (somewhere between one-third and two-thirds) relapsed into their previous mode of operation and results. I was concerned that although Kotter had been preaching "make it stick" for years, a disappointing number of organizations were not able to do it. Of course, most of them were not rigorously following Kotter's recommendations either.

Based on their research, the Heaths put forth six recommendations to make ideas stick:

1. Simplicity;
2. Unexpectedness;
3. Concreteness;
4. Credibility;
5. Emotions;
6. Stories.

It seems to me that all of these points align with what you have to do to create and communicate an effective project delivery process change vision. You will find that they follow their own advice and include many fascinating stories to make their ideas stick. I will not repeat any of it here, but I strongly recommend that you read their book before you embark on leading a CCPM implementation.

Then they published *Switch: How to Change Things When Change Is Hard* (Heath, 2010). It explicitly confirmed my belief that what they had researched and reported applied directly to organizational change management. They build their change approach using a model proposed by Johnathan Haidt in a book I had read earlier and also enjoyed: *The Happiness Hypothesis* (Haidt, 2006). Haidt is a psychologist at the University of Virginia and described a model of how the human brain works comparing it to the union of a human rider and an elephant. The little rider perched atop the enormous powerful elephant compares to your logical analytical thinking processes and the elephant compares to the vast and powerful computations going on below your awareness: the stuff of human behavior and emotions. They used this analogy to structure their organizational change model framework:

1. Direct the rider.
2. Motivate the elephant.
3. Shape the path.

Their detailed suggestions in each of these areas strongly support the directions given by Kotter, in particular emphasizing focusing on behaviors and engaging emotions (the elephants). They noted (p. 257), "If we really did understand why an extreme change like having kids works while minor changes routinely fail—if we really understood that change rarely happens unless it is motivated by feeling… managers would never kick off change initiatives with PowerPoint presentations." The Heaths strongly support Kotter's model and provide extensive supporting research and stories that can help you with your change.

I strongly recommend their books and Web site: www.Switchthebook.com/resources. They recently started a newsletter that I hope will last.

11.4.4 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 3 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 Super Bowl 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.

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.

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. Be sure to provide visible communications of how well your teams are doing.

11.4.5 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. However, 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 11.5 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

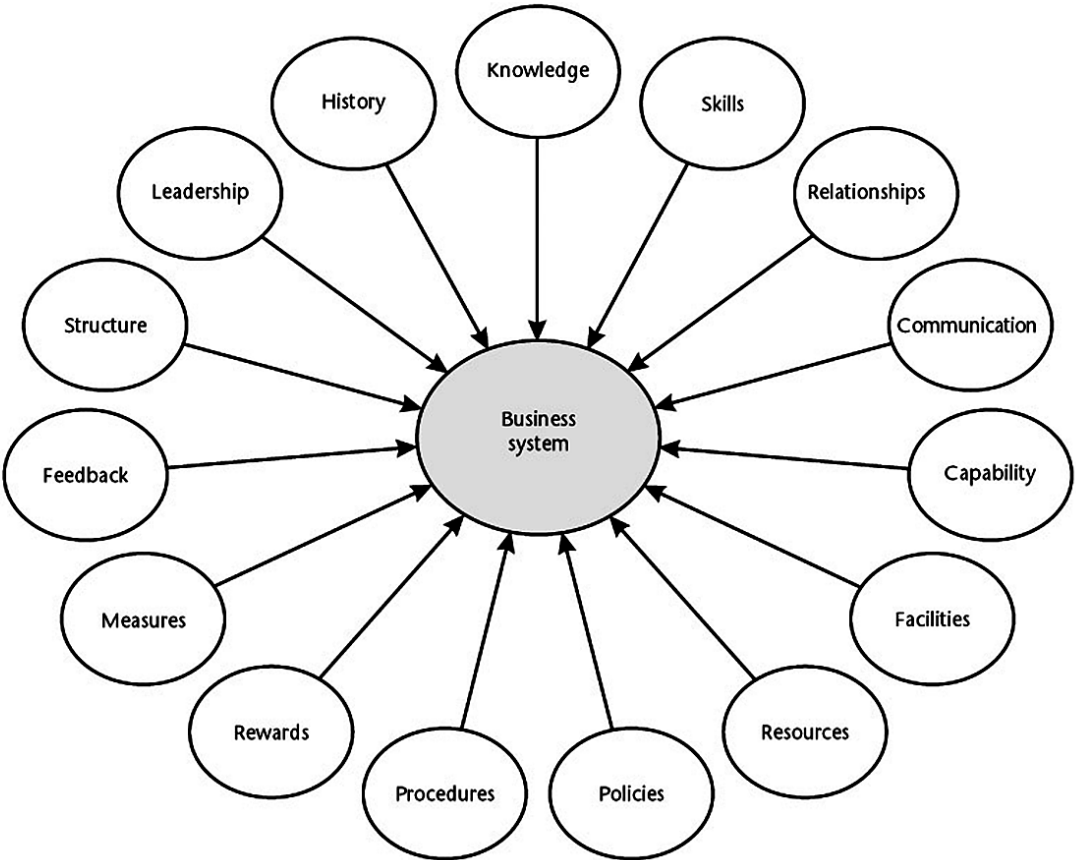


Figure 11.5 Business systems exist in a field of interrelated forces, which naturally push back on attempts to change the system.

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.

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.

11.5 The Need for Pilots

The most common response offered by organizations considering changing to CCPM is, “Let’s try a pilot project.” Although I originally discouraged it, I now heartily endorse it. Pilots represent an incremental approach to change. They reduce risk and increase the probability of long-term success. They are extremely useful to find out what has to be done to make CCPM succeed in your organization. Here are some considerations based on my experience with pilot projects:

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 project 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.
10. Celebrate successes, even small ones, and reinforce the behaviors that produced the successes.

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.

11.6 Example Objections

Listed next 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 have 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 be better than 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 have all the other companies in our business not implemented it?
2. We already use overly optimistic estimates and always work to meet or beat them.
3. It will never work here. CCPM is doomed to fail, given the inherent complexity of our work.
4. It will 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 will not 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 does not 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 or 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 do not, 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 are wasting our time. Let's get on with the real work.
16. What is so bad about the way we currently manage projects? If it's not broken, why are we trying to fix it?
17. And lastly, my personal favorite: We cannot implement something that will show a 50% to 100% improvement. It would mean that I have done a bad job my whole career.

Although I have a reservation about one aspect of the book, Kotter's most recent book, *Buy-In: Saving Your Good Idea from Getting Shot Down,* may help you deal with some kinds of resistance (Kotter, 2010). He provided strategies for dealing with 24 generic objections or obstructive tactics.

11.7 Ongoing Improvement

Once you have completed your first round of improving project delivery, you are staged to move on to improve what you have accomplished. Kotter's eighth step of sustainment is not enough. If you do not continue to improve, what you have accomplished will degrade over time as a result of a physical law known as the second law of thermodynamics: entropy (or chaos) ever increases. You cannot rest on your accomplishments.

I believe that the process you choose to lead ongoing improvement matters little as long as your management team commits to it as their job. Dr. W. Edwards Deming's first of 14 points was, "Create constancy of purpose toward improvement of product and service, with the aim to become competitive, stay in business and to provide jobs." TOC, Lean, and Six Sigma all offer an array of approaches and tools that will aid you in leading ongoing improvement.

I do not mean to discourage ongoing improvement in all workgroups with the methods suggested such as deploying Kanban or the other tools, but the greatest value to your organization will accrue in the shortest time if you follow the TOC five focusing steps to decide where to put Senior Management attention to apply those tools for continuous improvement. Each time you do an improvement cycle on a constraint, something else may become the constraint, leading to a process of ongoing improvement as described by Goldratt's fifth step: do not let inertia lead you to complacency with what you have accomplished.

I currently believe (subject to future improvement) that Rother presents a very fundamental point that many continuous improvement advocates have missed or at least do not emphasize enough (Rother, 2010). The world's most successful organization (Toyota) continues to succeed because of some deep-seated management practices that are so ingrained they are often overlooked: what he calls "katas." The two key katas he describes are continuous improvement and coaching. The continuous improvement kata means every manager believes that their first job is to continuously improve the production processes. That is what they were hired and then promoted to do. Other stuff must be subordinated to continuous improvement. The second kata is that the way to improve is through coaching your subordinates on improvement. This includes coaching them on how to coach.

Rother in his work described a process to lead improvement that goes beyond the scope of this book. I strongly recommend you read his book and follow up with accessing the information on his Web site as you move forward on your journey.

11.8 Summary

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:

1. Consider implementing Kanban at the work group level first. It teaches pull and implements WIP control at the working level which will reduce some of the multitasking waste and produce short-term wins.
2. Create an effective change plan to harness organizational dynamics to accelerate the change: create a sense of urgency and publicize short-term wins.
3. Senior Management must show visible leadership; they are the critical success factor for multiproject CCPM implementation.
4. Use your project process to implement the change to CCPM: charter, endorse, work plan, perform, and close.

5. Carefully define your pilot project(s), and support them.
6. Change management is becoming more and more of a well-researched discipline. Draw on the models developed by Kotter and Cohen, Prosci, and the Heath brothers to find your way.
7. Humans and organizations try to maintain equilibrium (i.e., appear to resist change). Your implementation plan should anticipate and plan for this. The resources provided in this chapter will help you do that.
8. Just do it.
9. And then never give up. Implement a process of ongoing improvement in which managers recognize leading work process improvement as the most important part of their jobs and see their job as coaching their subordinates to create and implement ongoing improvements.

Organizational understanding of TOC can greatly aid implementing CCPM but does not seem to be a necessary condition. Organizational project management skills are necessary. 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 in workgroups with Kanban and 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

Anderson, D. (2010). *KANBAN*. Sequim, WA: Blue Hole Press.

Anderson, D. (2011). *Kanban, When Is It Not Appropriate?* http://www.kanbaninaction.com.

Cohen, D. (2005). *The Heart of Change Field Guide*. Boston, MA: Harvard Business School Press.

Dalziel, Murray M., and Stephen C. Schoonover. (1988). *Changing Ways: A Practical Tool for Implementing Change Within Organizations*. New York: Amacom.

Daniels, A., and Daniels J. (2006) *Performance Management*. Atlanta, GA: Performance Management Publications.

Haidt, J. (2006). *The Happiness Hypothesis*. New York: Basic Books.

Heath, C., and Heath, D. (2008). *Made to Stick*. New York: Random House.

Heath, C., and Heath, D. (2010). *Switch: How to Change Things When Change Is Hard*. New York: Random House.

Hiatt, H. (2006). *ADKAR How to Implement Successful Change in Our Personal Lives and Professional Careers*. Loveland, CO: Prosci Research.

Hiatt, H., and Creasey, T. (2012). *Change Management*. Loveland, CO: Prosci, Inc.

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

Kotter, J., and Cohen, D. (2002). *The Heart of Change*. Boston, MA: Harvard Business School Press.

Kotter, J. (2010). *Buy-In: Saving Your Good Idea from Getting Shot Down*. Boston, MA: Harvard Business School Press.

Pirasteh, R., and Fox, R. (2011). *Profitability with No Boundaries*. Milwaukee, WI: ASQ Press.

PMI. (2013). *Managing Change in Organizations: A Practice Guide*. Newtown Square, PA: Project Management Institute.

Rother, M. (2010). *Toyota Kata*. New York: McGraw-Hill.

Sharot, Tali. (2011). *The Optimism Bias*. New York: Pantheon.

Glossary

Activity The lowest level of the WBS. A packet of work that forms the basic building block of a plan or network. Also called a task.

Activity network A network made up of two or more activities with dependency.

Actual cost (AC) The actual money spent in performing an activity so far.

Actual cost of work performed (ACWP) The earned value/cost schedule control system (CSCS) term for actual cost. Renamed actual cost (AC) by PMI.

Agile A style of project management that emerged as an approach to software development project management in the mid-1990s. Agile focuses on early delivery of business value using scope flexibility by quickly making adjustments to realities observed during actual project execution.

Bar chart In project management, a chart showing tasks with horizontal bars representing the duration of the task. See Gantt chart.

Baseline schedule The initial project schedule before project execution begins. Baseline schedules are one of the bases for project change control. They also are often contractual commitments.

Bias A systematic tendency to overestimate or underestimate point value estimates or for actual values to overrun or underrun point estimates for project schedules and budgets.

Body of knowledge All of the collected knowledge on a discipline represented by all publications on the topic.

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 approved cost for a project. The budget may include a cost buffer or a budget contingency or management reserve.

Budgeted cost of work performed (BCWP) Earned value of work done, equal to the amount that was budgeted for the activities completed. See Earned value (EV).

Budgeted cost of work scheduled (BCWS) The value of work that should have been completed by the current date according to the baseline plan. Renamed Plan Value (PV) by PMI.

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 management/buffer recovery Responding to schedule or cost buffer signals by following task priorities, limiting interruptions of active tasks, resolving issues quickly, or taking other situationally appropriate management actions to achieve project schedule and cost objectives.

Buffer penetration The amount of the buffer that has been used up by actual progress in the project.

Buffer trigger points Buffer penetration thresholds that provide guidance to project managers on when to begin planning for management control action and when to initiate the action.

Capacity constraint buffer Portion of constraint (drum) resource supply that is not committed to project schedules in a multiproject environment. It controls resource demand (limits WIP) to reduce project cycle time and to increase the Throughput of projects.

Capacity constraint resource In a multiproject environment, the resource that is used by all of the projects in the Pipeline and most often overloaded before Pipelining.

Categories of legitimate reservation (CLR) A set of logical tests for trees created by the thinking process.

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 not present unless there is another cause for the same effect. Causes may be single or may require other conditions to lead to the effect.

Change management Techniques used to facilitate the introduction and adoption of new processes and behavioral norms. Change management addresses human psychological and emotional needs as well as the mechanics of the new practices.

Charter See Project charter.

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 at least one assumption and, therefore, invalidate the arrow and dissolve the 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.

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.

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.

Conflict resolution diagram (CRD) An alternative title for the evaporating cloud.

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.

Contingency An added increment of time or budget intended to compensate for the uncertainty of an estimate and for variation from the expected value during execution.

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 schedule control system (CSCS or CS^2) 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 as the earned value: the measure of work completed.

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.

Cost variance (CV) The value of the work done less the actual cost of the work done (i.e., EV – AC). A negative number shows that 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 the erroneous belief that cost savings are additive.

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 leveling 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 The system of project schedule planning and execution deployed by Theory of Constraints (TOC) practitioners including the critical chain schedule, statusing and buffer management, and Pipelining for multiple projects but not including many or most features of professional project management as codified by the PMBOK.

Critical Chain feeding buffer (CCFB) A time buffer at the end of a project activity chain that feeds the critical chain.

Critical Chain schedule A late finish schedule controlled by the critical chain, including a critical-chain completion buffer (project buffer) and feeding 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, 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.

Critical Path Method (CPM) The original innovation in using networks and defining a critical path through the network.

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.

Data Any string of characters that describes something about our reality.

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.

Deterministic A term describing the belief that it is possible to make precise and accurate single-point estimates of task durations or costs.

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-Rope (DBR) A method for production scheduling. The Theory of Constraints-based production management system developed by Dr. Eliyahu Goldratt that served as a predecessor to his invention of Critical Chain. 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.

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 (EV) The value of the work done, where value is calculated in terms of the baseline cost. See Budgeted cost of work performed (BCWP).

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 fourth focusing step: a 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.

Escalation (of tasks) Promptly bringing issues to higher levels of management when the issues cannot be resolved at lower levels of management or by the performing resources.

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.

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 second focusing step: a 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).

Fever chart For schedules, a graph illustrating the percent of schedule buffer consumption relative to the percent completion of the critical chain. For budgets, a graph illustrating the rate of buffer cost consumption relative to percentage of budget spent.

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.

Fishbone diagram A problem-solving and brainstorming root cause analysis tool. Also known as a cause-and-effect diagram or an Ishikawa diagram.

Five Focusing Steps A five-step process to identify and increase the workflow through constraints.

Five S A system of workplace standardization and organization. Sort, Set in order, Shine, Standardize, and Sustain.

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.

Focus Resources work on one project or nonproject work task at a time and ask for management assistance if stuck.

Free float The amount of time an activity may be delayed without causing delay to successor activities.

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.

Gemba walk A visit to the physical location where work is performed with the purpose of understanding the real process that is used and obstacles to performing the work according to an established process standard.

Goal, The See Jonah.

Granularity A term referring to the degree of refinement of task sizing in a project schedule.

Heijunka Production leveling.

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.

Identify The first step of the TOC focusing process, consisting of identifying the system constraint.

Inertia The TOC fifth focusing step: do not let management inertia prevent a process of ongoing improvement.

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.

Kanban In Japanese, a sign board. A process developed by T. Ohno for just-in-time production processing. In this book a modified process to support local work in progress (WIP) control through visual control and pull processing.

Kata In Japanese martial arts, a learned form. In management, a cultural behavior of organization management.

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.

Lean A production management philosophy developed for the Toyota Production System that emphasizes removing waste. Lean also attempts to level unevenness of demand and to relieve overburden of the process (i.e., excessive work in process).

Level of effort Recurring tasks in a project schedule or tasks that extend for the duration of the schedule (e.g., project management).

Linked projects A term used in some computer packages to indicate projects that use a common set of resources.

Little's law An equation representing the relationship among the throughput of a system, the WIP, and the cycle time for individual projects through the system.

Logic link See Dependency links.

Logic loop A circular sequence of dependency links between activities in a network.

Master scheduler An organization's scheduling subject matter expert (SME). The master scheduler acts as the process owner for project scheduling and for developing and maintaining the project pipeline.

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. An activity of zero duration that represents a significant deliverable or stage of the project.

Milestone A significant accomplishment of a project short of project completion. Milestones do not require dates, but with CCPM if a date is applied to a milestone, it must be buffered.

Milestone plan A plan containing only milestones highlighting key points in the project.

Monte Carlo A computational algorithm that relies on repeated random sampling of task or cost probability distributions to obtain numerical results describing the range of expected overall project cycle time or cost.

Muda Waste in a process.

Mura Unevenness in process flow.

Muri Strain, either physical or mental, affecting work performance: overburden.

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 or nonproject activity at the same time.

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.)

Necessity tree A logic tree in which each item at the tail of an arrow must exist for the item at the head of the arrow to exist because of some assumption or obstacle represented by the arrow.

Need The requirement(s) that must be met to achieve an objective or goal.

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.

Nonworking time Activities such as vacation, holiday, sick leave, or training activities during which a resource is unavailable for productive work.

Obstacle An entity that prevents an effect from existing.

Operating expense All of the money that 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.

Parkinson's law The tendency for actual task cycle times to use the entire duration estimate made at the time of scheduling, sometimes stated as "The work expands to fill the available time."

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.

Pessimistic duration The longest of the three durations in the three-duration technique, or PERT.

Pipelining Demand leveling a project portfolio by sequencing project start dates in order to control the amount of work in progress (WIP) within predefined limits.

Plan A generic term used for a statement of intentions, whether they relate to time, cost, or quality in their many forms.

Planned value (PV) PMI's term for the budgeted cost of work schedule (BCWS).

Positive reinforcement Feedback that a performer views as favorable and which motivates the performer to increase the behavior that generated the feedback.

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 cause cannot 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 as: "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.

Problem A gap between what we want and what we have.

Process A sequence of interconnected activities, each of which has an input and an output.

Process behavior chart The voice of the process. A time series representation of sequential process results that characterizes the extent of common-cause variation in the process and which includes limits that indicate variation likely to be the result of special causes. Also called a control chart.

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 (PERT) A network-scheduling tool, initially distinguished from CPM by using three activity-duration cost 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 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 change control The process of identification of project scope and/or constraint changes as they occur, estimating project schedule and cost impacts, and obtaining approval of relevant stakeholders before implementing the change; that is, limiting changes during project execution.

Project charter A document issued by the project initiator or sponsor that formally authorizes the existence of a project and provides the project manager with the authority to apply organizational resources to project activities.

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 Management Body of Knowledge (PMBOK™) The entire body of knowledge describing project management. Distinguished from the Project Management Institute's document A Guide to the Project Management Body of Knowledge.

Project Management Office (PMO) An organization entity that sets the standards for project managers to follow in regard to project plans and scheduling.

Project manager The person appointed to take day-to-day responsibility for management of the project throughout all its stages.

Project (management) plan The document or collection of information on a Web server or intranet that describes how the project will be executed, monitored, and controlled. It includes the project scope, schedule, and budget.

Project risk management Identifying and quantifying risks, and planning and controlling response to risk.

Pull The behavior to produce a result only when the customer asks for it. Internal to an organization, the next process pulls in work to be performed only when the next process has available capacity. A key element of just-in-time processing and Kanban.

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."

Queue A quantity of people, objects, or tasks waiting for resources to become available so that processing can begin.

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 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.

Resource manager Manager responsible for providing the resources necessary to deliver quality work on project tasks.

Risk An uncertain event or condition that, if it occurs, has a positive or negative effect on one or more of the project objectives. Quantified as the combination of the probability and consequence of an outcome. Project risk affects outcomes relative to the project scope, cost, or schedule and may include safety, environment, business, and security risks.

Rolling wave planning A segmented, iterative approach to planning of long projects, planning project segments only as far into the future as project details can be known. As a project segment nears completion, the current knowledge and status form the basis for planning the subsequent project segment.

Root cause The cause that 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 The output of schedule model usually presenting linked activities with planned dates, task durations, milestones and required resources.

Schedule variance (SV) The value of the work done less the value of the work that should have been done (i.e., EV – PV). 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.

Senior manager Manager responsible for designing, leading, and improving the organization system to deliver on all customer needs.

Six Sigma A continuous improvement process that builds on the teachings of Dr. W. Edwards Deming relying on statistical methods.

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.

Stakeholder An individual, group, or organization who may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project.

Statement of work (SOW) A narrative description of the products, services, or results to be delivered by the project.

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. See also Hockey Stick.

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.

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 The standard methods, practices, and procedures for handling frequently occurring events within the project. Includes management approvals, controls, and technical requirements. Systems will also cover methods of handling information transfer and storage.

Task A term usually synonymous with activity.

Task manager Manager responsible and accountable for task completion. The task manager most often directly supervises the task performing resource.

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.

Theory of Knowledge A set of principles for understanding and interpreting new theories and for assessing their preference over existing theories.

Thinking process The five-step Theory of Constraints process that identifies what to change, what to change to, and how to cause the change.

Throughput All of the money that our customers pay us minus the raw material cost.

Toyota Production System Manufacturing principles used by Toyota that have been adopted in Western cultures as Lean Manufacturing.

Uncertainty The inability to precisely forecast the outcome of an activity.

Undesired effect (UDE) A negative entity that in some degree impedes the achievement of an organization's business goal. An undesired effect usually is the result of a cause-effect-cause chain originating in a root cause, root problem, or conflict.

Variance (statistical) A measure of the dispersion of a sample and estimate of the standard deviation of a population.

Variation The natural property of all measured quantities to vary with each repeated attempt to reproduce the same process output. Separation into two types; common-cause variation and special-cause variation, aids in process improvement.

Visual control Using displays in the workplace to visually drive desired behaviors.

Work breakdown structure (WBS) A tree diagram that breaks a project scope down into increasing levels of detail. Although the WBS numbering scheme may extend down to tasks, the work package is normally considered the lowest level of the WBS.

Work in progress (WIP) The amount of work going on within the organization. For projects at a high level, all of the projects that have been started and not finished. Also all of the project and nonproject tasks that have been started and are not finished.

Work package A group of dependent tasks to produce a major deliverable. Work packages normally range from one to 25 tasks and are frequently represented as a summary task in project schedule.

List of Acronyms

ABC	Antecedent behavior consequence
AC	Actual cost
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

EV	Earned value
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	Project Management Body of Knowledge
PMI	Project Management Institute
PMP	Project Management Professional
PRT	Prerequisite tree
PV	Planned value
R&D	Research and development
RDU	Remaining duration
RUP	Rational Unified Process
SEI	Software Engineering Institute
SIPOC	Supplier>Input>Process>Output>Customer
SOW	Statement 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 (Larry) Leach is the principal of Advanced Projects, Inc. (API), applying business tools to help clients improve their project delivery system. Prior to founding API, he was vice president of several Fortune 500 companies, managing projects up to one billion dollars (US). Leach has an M.S. in business management from the University of Idaho and in mechanical engineering from the University of Connecticut. Larry is certified by the Project Management Institute as a project management professional (PMP). He has authored papers and books, including *Critical Chain Project Management, Second Edition; Lean Project Management: Eight Principles for Success;* and *Lean Project Leadership,* as well as chapters in the *Wiley Guide to Project Management and Project Portfolio Management.* He has taught general management and project management classes for the MBA programs of the University of Phoenix and University of Idaho, for the Project Management Institute's seminars, and for many private companies.

Index

U

V