Fundamentals of
Project Management

Developing Core Competencies to Help Outperform the Competition

Second Edition

JAMES P. LEWIS

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional service. If legal advice or other expert assistance is required, the services of a competent professional person should be sought.

"PMI" and the PMI logo are service and trademarks registered in the United States and other nations; "PMP" and the PMP logo are certification marks registered in the United States and other nations; "PMBOK," "PM Network," and "PMI Today" are trademarks registered in the United States and other nations; and "Project Management Journal" and "Building professionalism in project management" are trademarks of the Project Management Institute, Inc.

Various names used by companies to distinguish their software and other products can be claimed as trademarks. AMACOM uses such names throughout this book for editorial purposes only, with no intention of trademark violation. All such software or product names are in initial capital letters or ALL CAPITAL letters. Individual companies should be contacted for complete information regarding trademarks and registration.

This book is dedicated to:

Those who gave their lives trying to save others in the terrorist attack on the World Trade Center; September 11,2001.

CONTENTS

FIGURE LIST

Fundamentals of Project Management

Although management of projects has been going on for thousands of years, the practice has been widely recognized as a discipline in its own right for only about ten years. Not only are there now numerous degree programs, but also some organizations have begun to ask their contractors to provide only project managers who have been certified as professional project managers by the Project Management Institute, the professional society for practitioners. They are also insisting that their own people become professional project managers.

In today's fast-paced world, those organizations that practice sound project management methods have a competitive advantage over those that fly by the seat of the pants. Why? Because competition is rapidly becoming *time-based,* as well as *cost-based.* That is, if you can get a product or service to market faster than anyone else, you have an edge on your competition. Furthermore, if you can control the costs of your work better than others, you can sell your products or services at lower margins, since "sloppy" management requires that goods be sold at higher margins in order to make sure the business is profitable.

What if you aren't dealing in products or services? The same

principle applies. If you are nonprofit or a government agency, you face competition from others who may be able to do your work more efficiently (and at lower cost). In short, we must all learn to work smarter, not harder, in order to survive in the twenty-first century. Managing projects better is one way to achieve that result.

In this book you will gain a fast-track approach to managing your own projects. You will learn the essential steps in setting up project plans, scheduling your work, and monitoring progress/ exercising control to achieve desired project results.

The approach outlined in this book is based on what is considered *best practice* by experts in the field. If you follow these methods, you will increase the probability that you can meet critical *performance, cost,* and *schedule* targets. Admittedly, there is a lot more to project management than can be presented in this short book, but if you learn the absolute essence of the tools, you can go on from there to increase your skill.

I have been involved in managing projects since the mid-1960s. I have taught the subject since 1983. So a large part of my life has been spent dealing with this subject. I am still learning. If you have suggestions, comments, or questions, please feel free to contact me at jlewis@lewisinstitute.com. And check out my Web site: www.lewisinstitute.com. I would enjoy hearing from you.

Jim Lewis
Vinton, Virginia
May 2001

An Overview of Project Management

W hat's all the fuss about, anyway? Since the first edition of this book was published in 1995, the Project Management Institute (PMI®), the professional organization for people who manage projects, has grown from a few thousand members to a membership of more than 75,000 persons. In 2000 its membership grew by about 30 percent.

A professional association? Just for project management? Isn't project management just a variant on general management?

Yes and no. There are a lot of similarities, but there are enough differences to warrant treating project management as a discipline separate from general management. For one thing, projects are more schedule-intensive than most of the activities that general managers handle. And the people in a project team often don't report directly to the project manager; they do report to most general managers.

So just what is project management, and, for that matter, what is a project?

A project is a multi-task job that has performance, cost, time, and scope requirements and that is done only one time.

To begin, a project is a multitask job that has performance, time, cost, and scope requirements and that is done only one time. If it is repetitive, it's not a project. A project *should* have definite starting and ending points (time), a budget (cost), a clearly defined scope-or magnitude-of work to be done, and specific performance requirements that must be met. I say "should" because seldom does a project conform to the desired definition. These constraints on a project, by the way, are referred to throughout this book as the PCTS targets.

> A **project is a problem scheduled for solution.**
> —J. M. Juran

Dr. J. M. Juran, the quality guru, also defines a project as a problem scheduled for solution. I like this definition because it reminds me that every project is conducted to solve some kind of problem for a company. However, I must caution that the word *problem* typically has a negative meaning, and projects deal with both positive and negative kinds of problems. For example, developing a new product is a problem, but a positive one, while an environmental cleanup project deals with a negative kind of problem.

Project Failures

The Standish Group (www.standishgroup.com) has found that only about 17 percent of all software projects done in the United States meet the original PCTS targets; 50 percent must have the targets changed—meaning that they are usually late and overspent and need to have the performance requirements reduced—and the remaining 33 percent are actually canceled. One year, the United States spent an aggregate of more than $250 billion on software development, so this means that $80 billion was completely lost on canceled projects. What is truly astonishing is that 83 percent of all software projects get into trouble!

The Standish study reported here was conducted in 1994. In the February 2001 issue of *Software Development* magazine, an

ad for a software development conference stated that U.S. companies spend about $140 billion on canceled and overbudget projects each year.

Now, lest you think I am picking on software companies, let me say that these statistics apply to many different kinds of projects. Product development, for example, shares similar dismal rates of failure, waste, and cancellation. Experts on product development estimate that about 30 percent of the cost to develop a new product is rework. That means that one of every three engineers assigned to a project is working full-time just redoing what two other engineers did wrong in the first place!

The reason for these failures is consistently found to be inadequate project planning. People adopt a *ready-fire-aim* approach in an effort to get a job done really fast and end up spending far more than necessary by reworking errors, recovering from diversions down blind alleys, and so on.

I am frequently asked how to justify formal project management to senior managers in companies, and I always cite these statistics. However, managers want to know whether using good project management really reduces the failures and the rework, and I can only say you will have to try it and see for yourself. If you can achieve levels of rework of only a few percent using a seat-of-the-pants approach to managing projects, then keep doing what you're doing! However, I don't believe you will find this to be possible.

The question I would ask is whether general management makes a difference. If we locked up all the managers in a company for a couple of months, would business continue at the same levels of performance, or would those levels decline? If they decline, then we could argue that management must have been doing something positive; if they improve, then management has clearly been getting in the way. I doubt that many general managers would want to say that what they do doesn't matter. However, we all know that there are effective and ineffective general managers, and this is true of project managers, as well.

What Is Project Management?

Project management is facilitating the planning, scheduling, and controlling of all activities that must be done to achieve project objectives. Those objectives include the PCTS targets mentioned previously. Notice that I say *facilitation* of planning. One mistake made by inexperienced project managers is to plan the project for the team. Not only do they then get no buy-in to their plan, but also the plan is usually full of holes. Managers can't think of everything, their estimates of task durations are wrong, and the entire thing falls apart after the project begins. The first rule of project management is that the people who must do the work should help plan it.

> **Project management is facilitating the planning, scheduling, and controlling of all activities that must be done to achieve project objectives.**

The role of the project manager is that of an enabler. Her job is to help the team members get the work completed, to "run interference" for them, to get scarce resources that they need, and to buffer them from outside forces that would disrupt the work. She is not a project czar. She should be—above everything—a *leader;* in the true sense of the word.

The best definition of leadership that I have found is the one by Vance Packard (1962). He says, "Leadership is the art of getting others to want to do something that you believe should be done." The operative word here is *want.* Dictators get others to do things that they want done. So do guards who supervise prison work teams. But a leader gets people to want to do the work, and that is a significant difference.

> **The first rule of project management is that the people who must do the work should help plan it.**

The planning, scheduling, and control of work is the manage-

ment or administrative part of the job. But without leadership, projects tend to just satisfy the bare minimum requirements. With leadership, they can exceed those bare minimums.

Not Just Scheduling!

One of the common misconceptions about project management is that it is just scheduling. At last report, Microsoft had sold more than a million copies of Microsoft Project, yet the failure rate remains high. Scheduling is certainly a major tool used to manage projects, but it is not nearly as important as developing a shared understanding of what the project is supposed to accomplish or constructing a good Work Breakdown Structure (WBS) to identify all the work to be done (I discuss the WBS later). In fact, without good project management, the only thing a detailed schedule is going to do is allow you to document your failures with great precision!

> **Leadership is the art of getting others to want to do something that you believe should be done.**
>
> —Vance Packard

I do want to make one point about scheduling software. It doesn't matter too much which package you select, as they all have strong and weak points. However, the tendency is to give people the software and expect them to learn how to use it without any training. This simply does not work. The features of scheduling software are such that most people don't learn the subtleties by themselves. They don't have the time, because they are trying to do their regular jobs, and not everyone is good at self-paced learning. You wouldn't hire an inexperienced person to run a complex machine in a factory and put him to work without training, because you would know he would destroy something or injure himself. So why do it with software?

One-Person Projects

When is managing a project not project management? When only one person is involved.

A lot of people are sent to my seminars to learn how to manage projects, even though they are the only person working on their projects. It is true that a one-person job can be called a project, because it has a definite starting point, target end date, specific performance requirements, defined scope of work, and budget. However, when no one else is working on the project (including outside vendors), there is no need for a critical path schedule. A critical path schedule is one that has a number of parallel paths, and one of them is longer than the others and determines how long it **will** take to complete the job, or, ultimately, whether the given end date can be met. When you're working on a job by yourself, there aren't any parallel paths—unless you are ambidextrous!

One-person projects do require good self-management, or good time management, but all you need is a good to-do list, which comes from a task listing. However, unless you are coordinating the work of other people, you aren't practicing true project management.

The Big Trap--Working Project Managers

It is common to have individuals serve as project managers and require that they do part of the actual work in the project. This is a certain prescription for problems. If it is a true team, consisting of several people, the project manager **will** inevitably find herself torn between managing and getting her part of the work done. Naturally, the work must take precedence, or the schedule will slip, so she opts to do the work. That means that the managing does not get done. She hopes it will take care of itself, but it never does. After all, if the team could manage itself, there would be no need for a project manager in the first place (remember our argument about whether project management matters?).

Unfortunately, when time comes for her performance evaluation, she **will** be told that her managing needs improving. Actually, she just needs to be allowed to practice management in the first place.

For very small teams—perhaps up to three or four people—a

project manager can do some of the work. But, as team sizes increase, it becomes impossible to both work and manage, because you are constantly being pulled away from the work by the needs of your team members.

One reason for this situation is that organizations don't fully understand what project management is all about; they think that it is possible for individuals to do both. The result is that nearly everyone in the company is trying to manage projects; as is true in every discipline, some managers will be good at it, and others will have no aptitude whatsoever. I have found that a far better approach is to select a few individuals who have the aptitude and the desire to be project managers and to let them manage a number of small projects. This frees "technical" people (to use the term broadly) to do technical work without having to worry about administrative issues and allows project managers to get really good at their jobs.

It is outside the scope of this book to discuss how to select project managers, but, for the interested reader, the topic is covered in *The World-Class Project Manager* (Wysocki & Lewis, 2001).

You Can't Have It All!

One of the common causes of project failures is that the project sponsor demands that the project manager finish the job by a certain time, within budget, and at a given magnitude or scope, while achieving specific performance levels. In other words, the sponsor dictates all four of the project constraints. This doesn't work.

The relationship between the PCTS constraints can be written as follows:

$$C = f(P, T, S)$$

In words, this says, "Cost is a function of Performance, Time, and Scope. Graphically, I like to show it as a triangle, in which P, C, and T are the sides and S is the area. This is shown in Figure 1-1.

In geometry, we know that if we are given values for the

**Figure 1-1. Triangles showing the relationship
between P, C, T, and S.**
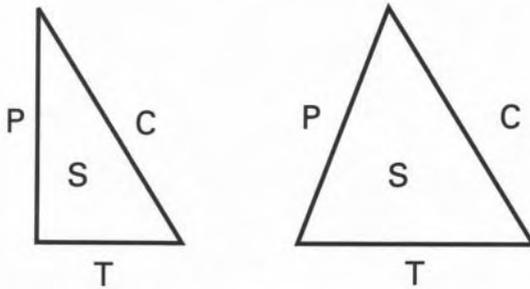
sides of a triangle, we can compute the area. Or, if we know the area and the lengths of two sides, we can compute the length of the remaining side. This translates into a very practical rule of project management: The sponsor can assign values to any three variables, but the project manager must determine the remaining one.

So let's assume that the sponsor requires certain performance, time, and scope from the project. It is the project manager's job to determine what it will cost to achieve those results. However, I always caution project managers that they should have a paramedic standing by when they give the cost figure to the sponsor because she **will** probably have a stroke or heart attack, and the paramedic will have to revive her.

Invariably, the sponsor will exclaim, "How can it cost that much?" She has a figure in mind, and your number will always exceed her figure. She may say, "If it's going to cost that much, we can't justify doing the job." Exactly! And that is the decision she should make. But she is certain to try to get you to commit to a lower number, and, if you do, then you only set up yourself—and her—to take a big fall later on.

It is your *obligation* to give the sponsor a valid cost so that she can make a valid decision about whether the project should be done. If you allow yourself to be intimidated into committing to a lower number, you **will** face a disaster later on, and you are far better off taking your lumps now than being hung later on.

Of course, there is another possibility. If she says she can afford only so much for the job, then you can offer to reduce the scope. If the job is viable at that scope level, then the project can be done. Otherwise, it is prudent to forget this project and do something else that can make a profit for the company. As someone has said, there is a higher probability that things will accidentally go wrong in a project than that they will accidentally go right. In terms of cost estimates, this means that there is always a higher likelihood that the budget will be overrun than that the project will come in below budget. Just another way of stating Murphy's law, that "whatever can go wrong **will** go wrong."

> There is a higher probability that things will accidentally go wrong in a project than that they will accidentally go right.

The Phases of a Project

There are many different models for the phases a project goes through during its life cycle. One model captures the all-too-frequent nature of projects that are not managed well; this is shown in Figure 1-2.

I have shown this diagram to people all over the world, and they invariably laugh and say, "Yes, that's the way it works." I suppose I can take comfort that we Americans are not the only ones who face the problem, but the bad news is that there must be a lot of dysfunctional projects if everyone recognizes the model.

The appropriate life cycle model is shown in Figure 1-3. Notice that every project begins as a concept and that the project team must formalize the definition of the job before doing any work. However, with our ready-fire-aim mentality, we often start working on the job without ensuring that we have a proper definition or that everyone shares the mission and vision for the job.

Figure 1-2. Life cycle of a troubled project.
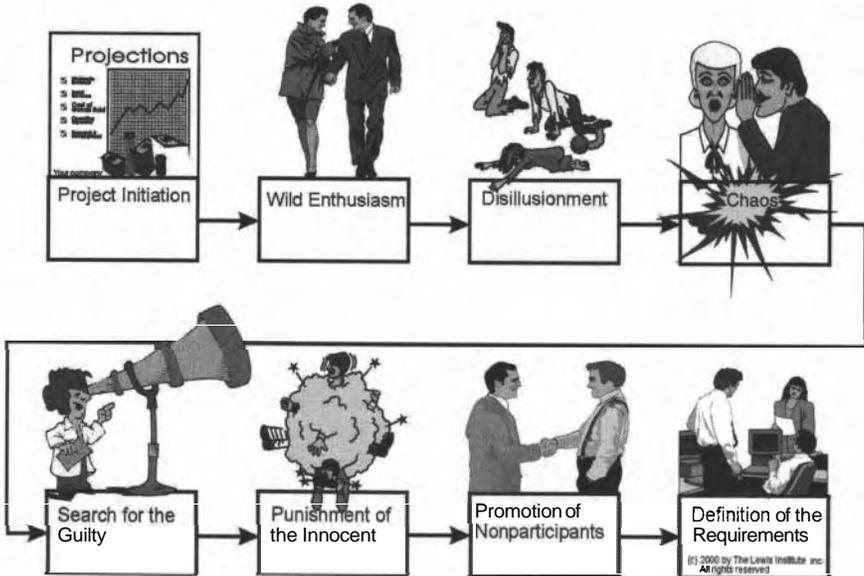
Definition Phase

Some years ago, a project manager in one of my client companies called me and said, "I've just had a conference call with key members of my project team, and I realized that we don't agree on what the project is supposed to accomplish."

I assured him that this was common.

"What should I do?" he asked.

I told him that he had no choice but to get the team members all going in the same direction by clarifying the mission of the project. He asked me to facilitate a meeting to do this. At the meeting, I stood in front of a flip chart and began by saying, "Let's write a problem statement." Someone immediately countered by saying, "We don't need to do that. We all know what the problem is."

I was unmoved by this comment. I said, "Well, if that is true, it's just a formality and will only take a few minutes, and it would help me if we wrote it down, so someone help me get started."

Figure 1-3. Appropriate project life cycle.
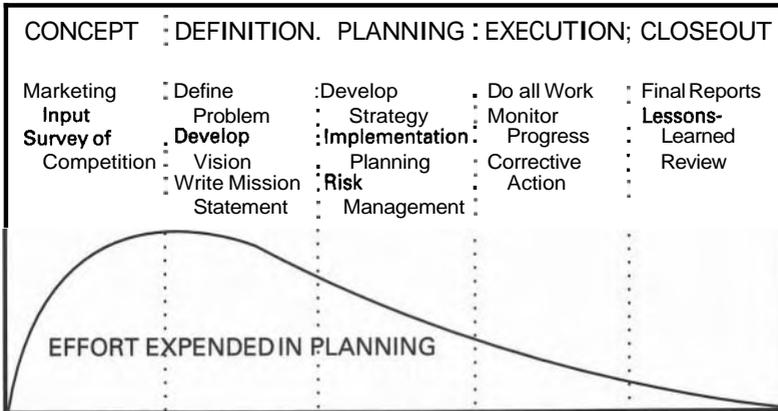
I'm going to be a little facetious to illustrate what happened next. Someone said, "The," and I wrote the word on the chart, and someone else said, "I don't agree with that!"

Three hours later we finally finished writing a problem statement.

The project manager was right. They did not agree on what the problem was, much less how to solve it. This is fundamental— and is so often true that I begin to think we all have a defective gene that prohibits us from insisting that we have a good definition of the problem before we start the work. Remember, project management is solving a problem on a large scale, and the way you define a problem determines how you **will** solve it. If you have the wrong definition, you may come up with the right solution—to the wrong problem!

In fact, I have become convinced that projects seldom fail at the end. Rather, they fail at the definition stage. I call these projects *headless-chicken projects* because they are like the chicken that has had its head chopped off and runs around spewing blood everywhere before it finally falls over and is "officially" dead. Projects work the same way. They spew blood all over the place, until someone finally says, "I think that project is dead," and indeed it is. But it was actually dead when its

head was chopped off in the beginning — it just took a while for everyone to realize it.

Once the project is defined, you can plan how to do the work. There are three components to the plan: strategy, tactics, and logistics. Strategy is the overall approach or "game plan" that will be followed to do the work. An example of strategy was related to me by a friend who is into military history.

Strategy

During World War II, defense contractors were under great pressure to build weaponry at an intense level. To accelerate construction of ships and planes in particular, many new assembly methods were invented. Avondale shipyards, for example, worked on finding better ways to build ships. The traditional way had always been to build the ship in an upright position. However, ships built from steel required welding in the bottom, or keel area of the boat, and this was very difficult to do in the traditional model. Avondale decided to build its ships upside down, to make the welding easier, and then to turn them over to complete the structures above the top deck. This strategy was so effective that it could build boats faster, cheaper, and of higher quality than its competitors, and the strategy is still being used today, nearly sixty years later.

Implementation Planning

This phase includes tactics and logistics. If you are going to build ships upside down, you must work out the details of how to do it. A fixture must be constructed that **will** hold the ship and allow it to be turned over without being damaged. This step is called working out the tactics and also includes determining the sequence in which the work will be done, who will do what, and how long each step will take.

Logistics deals with making sure the team has the materials and other supplies needed to do its job. Ordinarily we think about providing teams with the raw materials they need, but if

the project is in a location where people can't get food, work will soon come to a grinding halt. So provisions must be made for the team to be fed—and possibly housed.

Execution and Control

Once the plan has been developed and approved, the team can begin work. This is the execution phase. This phase also includes control; while the plan is being implemented, someone must ensure that the work is progressing according to the plan. When deviations from the plan occur, corrective action must be taken to get the project back on track; if this is not possible, the plan must be changed and approved. The revised plan then becomes the new baseline against which progress is tracked.

Closeout

When all the work has been completed, the closeout phase requires that a review of the project be conducted. The purpose is to learn lessons from this job that can be applied to future ones. Two questions are asked: "What did we do well?" and "What do we want to improve next time?"

Notice that we don't ask what was done wrong. This question tends to make people defensive, and they try to hide things that may result in their being punished. In fact, a lessons-learned review should never be conducted in a blame-and-punishment mode. If you are trying to conduct an inquisition, that's different. The purpose of an inquisition is usually to find who is responsible for major disasters and then to punish them. Lessons learned should mean exactly what the words imply.

I have learned during the past few years that very few organizations do regular lessons-learned reviews of their projects. There is a reluctance to "open a can of worms." And there is a desire to get on with the next job. The problem is that you are almost sure to repeat the mistakes made on the previous project if no one knows about them or has an understanding of how they happened so that they can determine how to prevent them. But,

perhaps most important, you can't even take advantage of the good things you did if you don't know about them.

It has been said that the organizations that **will** survive and thrive in the future will be those that learn to learn faster than their competitors. This seems especially true for projects.

The Steps in Managing a Project

The actual steps to manage a project are straightforward. Accomplishing them may not be. The model in Figure 1-4 illustrates the steps.

Subsequent chapters of this book elaborate on how each step is accomplished. For now, here is a brief description of the actions involved.

DEFINE THE PROBLEM

As was discussed previously, you need to identify the problem to be solved by the project. It helps to visualize the desired end result. What will be different? What will you see, hear, taste, touch, or smell? (Use sensory evidence if things can't be quantified.) What client *need* is being satisfied by the project?

DEVELOP SOLUTION OPTIONS

How many different ways might you go about solving the problem? Brainstorm solution alternatives (you can do this alone or as a group). Of the available alternatives, which do you think will best solve the problem? Is it more or less costly than other suitable choices? Will it result in a complete or only a partial fix?

PLAN THE PROJECT

Planning is answering questions—what must be done, by whom, for how much, how, when, and so on. Naturally, answering these questions often requires a crystal ball. We discuss these steps in more detail in chapters 2 through 4.

Figure 1-4. The steps in managing a project.

```
┌──────────────────────────────┐
│      Define  the Problem      │
└──────────────────────────────┘
               │
               ▼
┌──────────────────────────────┐
│    Develop  Solution Options   │
└──────────────────────────────┘
               │
               ▼
┌──────────────────────────────┐
│        Plan  the Project       │
│  What must be done?            │
│  Who will do it?               │
│  How will it be done?          │
│  When must it be done?         │
│  How much will it cost?        │
│  What do we need to do it?     │
└──────────────────────────────┘
               │
               ▼
┌──────────────────────────────┐
│        Execute the Plan        │
└──────────────────────────────┘
               │
               ▼
┌──────────────────────────────┐
│   Monitor & Control Progress   │
│  Are we on target?             │
│  If not, what must be done?    │
│  Should the plan be changed?   │
└──────────────────────────────┘


┌──────────────────────────────┐
│         Close  Project         │
│  What was done well?           │
│  What should be improved?      │
│  What else did we learn?       │
└──────────────────────────────┘
```
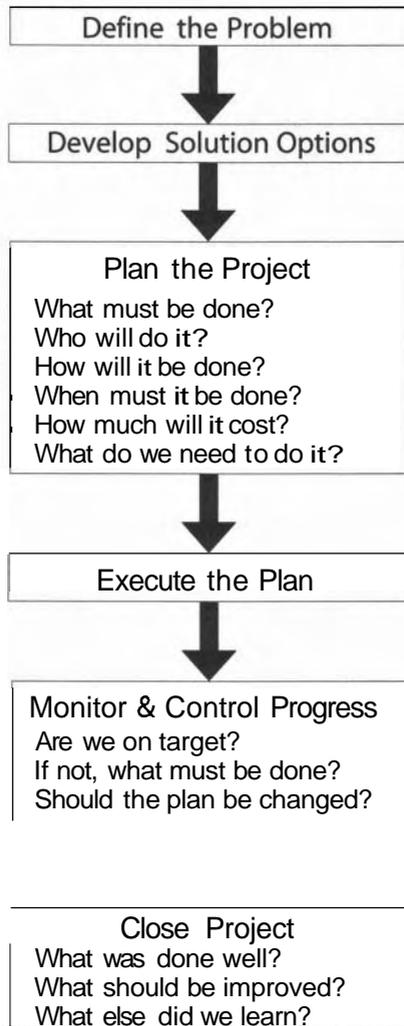
EXECUTE THE PLAN

Obvious. Once the plan is drafted, it must be implemented. Inter-
estingly, we sometimes find that people go to great effort to put
together a plan, then fail to follow it. If a plan is not followed,
there is not much point in planning, is there?

MONITOR AND CONTROL PROGRESS

Plans are developed so that you can achieve your end result suc-cessfully. Unless you monitor your progress, you cannot be sure you will succeed. It would be like having a road map to a desti-nation but not monitoring the highway signs.

Of course, if a deviation from the plan is discovered, you must ask what must be done to get back on track, or—if that seems impossible—how the plan should be modified to reflect the new realities.

CLOSE THE PROJECT

Once the destination has been reached, the project is finished, but you should take one final step. Some people call it an audit, others a postmortem (sounds a bit morbid, doesn't it?). Whatever you call it, the point is to learn something from what you just did. Note the way the questions are phrased: What was done well? What should be improved? What else did we learn? We can *always* improve on what we have done. However, asking What did we do wrong? is likely to make people a bit defensive, so the focus should be on improvement, not on placing blame. More on this later.

The Project Management Body of Knowledge (PMBOK®)

The Project Management Institute has attempted to determine a minimum body of knowledge that a project manager needs in order to be effective. At present, PMI identifies nine general areas of knowledge, which I summarize here. If you want a complete document, you can get one by visiting the PMI Web site: www.pmi.org.

The nine areas of knowledge are these:

1. Project integration management. Project integration man-agement ensures that the project is properly planned, executed, and controlled. It includes the exercise of formal project *change* control.

2. Project scope management. Changes to project scope are often the factors that kill a project. Scope management includes authorizing the job, developing a scope statement that defines the boundaries of the project, subdividing the work into manageable components with deliverables, verifying that scope planned has been achieved, and implementing scope change control procedures.

3. *Project time management.* I consider this a bad choice of terms, since time management implies personal efforts to manage one's time. For projects, this amounts to developing a schedule that can be met, then controlling work to ensure that it is! It's that simple.

4. Project cost management. This is exactly what it sounds like. It involves estimating the cost of resources, including people, equipment, materials, and items such as travel and other support details. After this is done, costs are budgeted and tracked to keep the project within that budget.

5. Project quality management. **As** I have commented earlier, one cause of project failure is the tendency to overlook or sacrifice quality in order to meet a tight deadline. It is not very helpful to complete a project on time, only to discover that the thing delivered won't work properly! Quality management includes both quality assurance (planning to meet quality requirements) and quality control (steps taken to monitor results to see whether they conform to requirements).

6. Project human resource management. Managing human resources is often overlooked in projects. It involves identifying the people needed to do the job; defining their roles, responsibilities, and reporting relationships; acquiring those people; and then managing them as the project is executed.

7. Project communications management. As the title implies, communication management involves planning, executing, and controlling the acquisition and dissemination of all information relevant to the needs of all project stakeholders. This

information includes project status, accomplishments, events that may affect other stakeholders or projects, and so on.

8. Project risk management. Risk management is the systematic process of identifying, analyzing, and responding to project risk. It includes maximizing the probability and consequences of positive events and minimizing the probability and consequences of adverse events to project objectives.

9. Project procurement management. Procurement of necessary goods and services for the project is the *logistics* aspect of managing a job. It involves deciding what must be procured, issuing requests for bids or quotations, selecting vendors, administering contracts, and closing them when the job is finished.

● ●
Key Points to Remember

A project is a one-time, multitask job that has definite starting and ending points, a well-defined scope of work, a budget, and a temporary team that will be disbanded once the job is finished.

■ A project is also a problem scheduled for solution.

Project management is facilitation of the planning, scheduling, and controlling of all activities that must be done to achieve project objectives.

All projects are constrained by Performance, Time, Cost, and Scope requirements. Only three of these can have values assigned. The fourth must be determined by the project team.

Projects tend to fail because teams do not take time to ensure that they have developed a proper definition of the problem being solved.

■ The major phases of a project include concept, definition, planning, execution and control, and closeout.
● ●

Qurtionr for Review . . . .

1. Project management is not just:
   a. Planning
   b. Rework
   c. Scheduling
   d. Controlling
2. The problem with being a working project manager is that, in a conflict between working and managing:
   a. You won't know what priorities to set.
   b. Your boss will think you're slacking off.
   c. There will never be enough time to do both.
   d. The work will take precedence, and managing will suffer.
3. PMBOK defines:
   a. The body of knowledge identified by PMI that project managers need to be effective
   b. A test administered by PMI to certify project managers
   c. An acronym for a special kind of risk analysis, like FMEA
   d. None of the above
4. Project scope defines:
   a. A project manager's visibility to the end date
   b. The magnitude or size of the job
   c. How often a project has been changed
   d. The limits of a project manager's authority

Planning the Project

n chapter 1 I talked about the high cost of project failures. Almost every study finds that failures are caused primarily by poor project management, especially the failure to plan properly. There are two barriers to good planning. The first is prevailing paradigms, and the second is the nature of human beings.

A paradigm is a belief about what the world is like. You can tell what people believe by watching what they do, because they always behave consistently with their deeply held beliefs. It is not necessarily what they *say* they believe, but what they *really* believe that counts. Chris Argyris has called these beliefs one's *theory espoused* as opposed to one's *theory in practice. (Overcoming Organizational Defenses, 1990).*

To illustrate, a fellow who attended my seminar on the tools of project management later told me that, upon returning to work, he immediately convened a meeting of his project team to prepare a plan. His boss called him out of the conference room.

"What are you doing?" asked the boss.

"Planning our project," explained the fellow.

"Oh, you don't have time for that nonsense," his boss told him. "Get them out of the conference room so they can get the job done!"

It is clear that his boss didn't believe in planning, which raises the question, Why did he send the fellow to a training program if he really didn't believe in what is taught? Go figure.

The second reason that people don't plan is that they find the activity painful. Some individuals, especially engineers and programmers, are concerned that they will be held to estimates of task durations that they have made using their best guesses. Because they have no historical data to draw on, this is all they can do. But they also know that such numbers are highly uncertain, and they are afraid that failure to meet established targets will get them in trouble. As one of my engineers told me once, "You can't schedule creativity."

I replied that this may be true, but we must pretend we can, because no one will fund the project unless we put down a time. Since then, I have changed my mind—you can schedule creativity, within limits. In fact, there is no better stimulus to creative thinking than a tight deadline. If you give people forever, they simply mess around and don't produce anything.

Nevertheless, we find that, when people are required to plan a project, they find the activity painful, and they resist the pain it causes. The net result is that they wind up on the pain curve numbered 1 in Figure 2-1. The net result of being on this curve is to experience a lot of pain, because the total pain experienced is represented by the area under the curve.

In curve 2 of the figure, there is a lot of pain early on, but it diminishes over time, and the total area under the curve is less than that under curve 1.

The Absolute Imperative of Planning

If you consider the major function of managing, it is to ensure that desired organization objectives are met. This is accomplished by exercising control over scarce resources. However,

Figure 2-1. Two pain curves in a project over time.
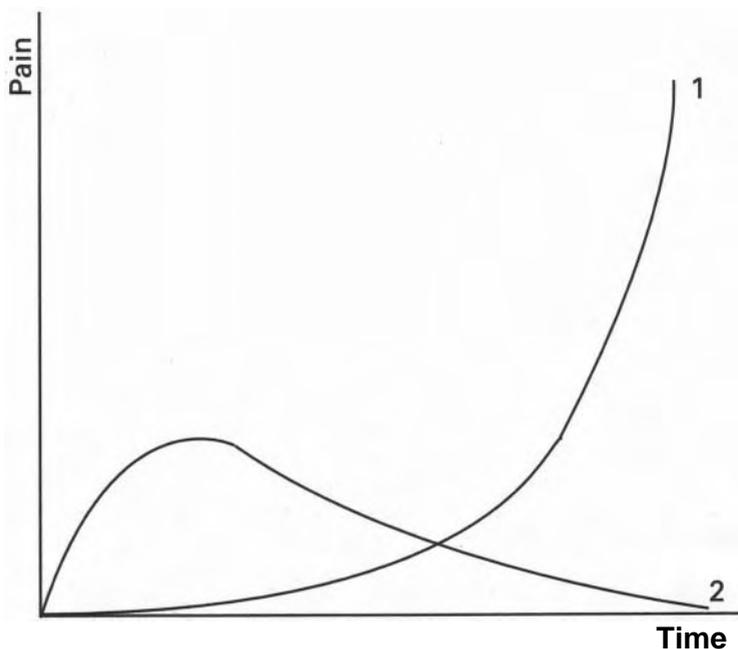
the word control has two connotations, and we must be careful which one we intend.

One meaning of the word is power and domination. In management, this is sometimes called the command-and-control approach, which in its worst form degenerates into the use of fear and intimidation to get things done. This method works when people have no other desirable options for employment or are not free to leave (as in the military or a prison). However, in a robust economy, very few will tolerate such management for long.

The second meaning of control—and

Control is exercised by comparing where you are to where you are supposed to be so that corrective action can be taken when there is a deviation.

the one I advocate for managers—is highlighted in the extracted quotation. Control is exercised by comparing where you are to where you are supposed to be so that corrective action can be taken when there is a deviation. Notice that this is an information systems or **| No plan, no control!** guidance definition. Further note that two things are necessary for control to exist. First, you must have a plan that tells where you are supposed to be in the first place. If you have no plan, then, you cannot possibly have control. I think we need to remind ourselves of this almost every day, because it is so easy to forget when you are constantly being assaulted by demands to do this and that and a million other things.

Predicting the future

is easy. It's knowing what's going on now

Second, if you don't know where you are, you can't have control. **As** the second quotation says, knowing where you are isn't as easy as it may seem, especially in doing knowledge work.

—Fritz R. S. Dressler

For example, you say you expect to write 10,000 lines of code by today, and you've written 8,000. Does that mean you're 80 percent of where you should be? Not necessarily. You may have found a more efficient way to write the code.

In any event, the major point to remember is that you cannot have control unless you have a plan, so planning is not an option.

Another trap that causes people not to plan is to believe that they have no time to plan; they need to get the job done really fast! This is counterintuitive, but think about it—if you have forever to get something done, then you don't need a plan. It's when the deadline is tight that the plan becomes really important. **As** a simple example, imagine flying into Chicago and being late. You have a meeting across town in less than an hour. You've never been to Chicago, but when the rental car attendant asks if you need a map, you say, "I don't have time for a map. I've got to get to my meeting really fast!" Not very likely, is it?

Planning Defined

Planning is quite simply answering the questions shown in Figure 2-2. They may be called the "Who, What, When, Why, How Much, How Long?" questions that you learned if you ever studied interviewing methods. It is that simple. And it is that hard. I say hard because answering some of these questions requires a crystal ball—especially questions like "How long will that take?" On tasks for which no history is available, this is a very hard question to answer. **As** my engineer said, "You can't schedule creativity."

Figure 2-2. Planning is answering questions.

Strategy, Tactics, and Logistics

To plan a project properly, you must attend to three kinds of activities that may have to be performed during the life of the job. These are strategy, tactics, and logistics.

Strategy refers to the overall method you will employ to do the job, sometimes referred to as a "game plan." **As** I related in Chapter 1, for thousands of years boats have been built with the keel down so that when one wishes to put the boat in the water, it is already right side up. This method worked fine until the

1940s, when World War II placed tremendous pressure on ship-yards to build military ships faster, and ships were being built out of steel plate, rather than wood. Shipbuilders quickly found that it was extremely difficult to weld in the keel area. From the out-side, you had problems getting under the ship, and inside you had to stand on your head to weld.

Avondale shipyards decided that it would be easier to build steel boats if they built them upside down. The welding in the keel area now could be done from outside, standing above the ship, and to work on the inside one could stand upright. This strategy proved so effective that Avondale could build boats faster, cheaper, and of higher quality than its competitors, and the approach is still being used today.

Too often planners choose a project strategy because "it has always been done that way," rather than because it is best. You should always ask yourself, "What would be the best way to go about this?" before you proceed to do detailed implementation planning.

Implementation Planning

Once you have decided to build boats upside down, you must work out all of the details of how it will be done. Sometimes we say that we must be sure to dot all of the "i's" and cross all the "t's." This is where you answer those "who, what, when, and where" questions. In fact, it is implementation planning that many people think of when they talk about planning. However, a well-developed implementation plan for the wrong project strat-egy can only help you fail more efficiently.

Logistics

Military people can quickly tell you the benefit of attention to logistics. You can't fight a battle if people have no ammunition, food, clothing, or transportation. It is logistics that attends to these things. I once saw a project scheduling program (regrettably now defunct) that allowed construction managers to record when a certain quantity of bricks was delivered to their site; it

then showed when they would run out, given a specific utilization rate. This would alert managers to schedule delivery of a new supply just before the existing stock was depleted.

I was also told about a road construction project in India that had very bad living conditions for the workers. The food was bad, sleeping conditions were poor, and the workers were suffering low morale. The project manager and his staff were all staying in a nice hotel in the nearby city. They finally realized the problem and moved to the site with the workers. Living conditions immediately improved, and so did worker morale. This is an example of the importance of a peripheral aspect of logistics.

Plan Ingredients

Following are the *minimum ingredients* that should be contained in a project plan. It is a good idea to keep these in a loose-leaf notebook. Initially, the notebook will contain only the plan. As the project is managed, reports, changes, and other documents **will** be added, so that when the project is completed the notebook **will** contain a complete history of the project, which can be used by others as data for planning and managing their own projects.

A suggestion here—until recent years, project notebooks have been the only way to completely document a project. With the advent of good personal computer databases, I suggest that the notebook be backed up with electronic data. It is very difficult to locate data in a notebook. Transferring information to a computer database makes it much easier to access. This may be the *only* device used a few years from now.

In any case, here are the items that make up the project plan:

- Problem statement.

- Project mission statement (see chapter **3** for instructions on how to develop a mission statement).

- Project   objectives (see chapter **3).**

- Project work requirements, including a list of all *deliverables,*

such as reports, hardware, software, and so on. It is a good idea to have a deliverable at each major project milestone so that progress can be measured more easily.

■ Exit criteria. Each milestone should have criteria established that will be used to determine whether the preceding phase of work is actually finished. If no deliverable is provided at a milestone, then exit criteria become very important.

■ End-item specifications to be met: This means engineering specifications, architectural specs, building codes, government regulations, and so on.

► Work Breakdown Structure (WBS). This is an identification of all of the tasks that must be performed in order to achieve project objectives. A WBS is also a good graphic portrayal of project *scope* (see chapter 5).

➥ Schedules (both milestone and working schedules should be provided; see chapters 5 and 6).

➥ Required resources (people, equipment, materials, and facilities). These must be specified in conjunction with the schedule (see chapters 4 and 5).

■ Control system (see chapters 7 and 8).

► Major contributors. Use a Linear Responsibility Chart for this (see chapter 4).

➥ Risk areas with contingencies when possible (see chapter **3).**

Sign-Off of the Plan

Once the plan has been prepared, it should be submitted to *stake-holders* for their signatures.

Following are some comments about the meaning of a signature and suggestions for handling the process:

► A signature means that the individual is *committed* to his contribution, agrees with the scope of work to be done, and

accepts the specs as valid. A signature on the part of a contributor does not mean a *guarantee* of performance. It is a commitment. Because there are factors outside our control, few of us would like to guarantee our performance. However, most would be willing to make a commitment, meaning we promise to do our best to fulfill our obligations. If a signature is treated as a guarantee, either signers will refuse to sign or they will sign without feeling really committed to the agreement. Neither response is desirable.

STAKEHOLDER: Anyone who has a vested interest in the project. These include contributors, customers, managers, and financial people.

▶ The plan should be signed in a *project plan review meeting,* not by mail. Circulating copies for signature by mail seldom works, as people may be too busy to read in depth and may miss important points that would be brought out in a signoff meeting.

The project plan should be reviewed and signed off in a meeting—not through interoffice mail!

■ People should be encouraged to "shoot holes in the plan" during the review meeting, rather than wait until problems develop later on. Naturally, this does not mean that they should nitpick the plan. The objective is to ensure that the plan is workable; that is all.

Changing the Plan

It would be nice to think that a plan, once developed, would never change. However, that is unrealistic. No one has 20/20 foresight. Unforeseen problems are almost certain to arise. The important thing is to

Encourage people to spot problems during the sign-off meeting, not later.

make changes in an orderly way, following a standard change procedure.

If no change control is exercised, the project may wind up over budget, behind schedule, and hopelessly inadequate, with no warning until it is too late. Here are suggestions for handling changes to the plan:

> **Make changes in an orderly way, follow-ing a standard change procedure.**

■ Changes should be made only when a significant deviation occurs. A significant change is usually specified in terms of percent tolerances relative to the original targets.

▪ Change control is necessary to protect *everyone* from the effects of scope creep — changes to the project that result in additional work. If changes in scope are not identified and managed properly, the project may come in considerably over budget and/or behind schedule.

> **Any plan is bad which is not susceptible to change.**
>
> —Bartolommno de San Concordio (1475–1517)

.Causes of changes should be documented for reference in planning future projects. The causes should be factual, not blame-and-punishment statements.

Suggestions for Effective Planning

Here are some ideas to help you plan effectively:

Plan to plan. It is always difficult to get people together to develop a plan. The planning session itself should be planned or it may turn into a totally disorganized meeting of the type that plagues many organizations. This means that an agenda must be prepared, the meeting should be time-limited to the degree possible, and people should be kept on track. That is, if someone goes off on a tangent, the meeting facilitator should get the person back on track as

quickly as possible. There are many excellent guides to running meetings (e.g., *Mining Group Gold*, by Tom Kayser [Kayser, 1995]); the reader is referred to those.

The people who must implement a plan should participate in preparing it. Otherwise, you risk having contributors who feel no sense of commitment to the plan; their estimates may be erroneous, and major tasks may be forgotten.

Rule: The people who must do the work should participate in developing the plan.

The first rule of planning is to be prepared to replan. Unexpected obstacles will undoubtedly crop up and must be handled. This also means that you should not plan in too much detail if there is a likelihood that the plan **will** have to be changed, as this wastes time.

Because unexpected obstacles will crop up, always conduct a risk analysis to anticipate the most likely ones. Develop Plan B just in case Plan A doesn't work. Why not just use Plan B in the first place?

The first rule of planning is to be prepared to replan!

Because Plan **A** is better but has a few weaknesses. Plan B has weaknesses, also, but they must be different from those in Plan **A,** or there is no use in considering Plan B a backup.

The simple way to do a risk analysis is to ask, "What could go wrong?" This should be done for the schedule, work performance, and other parts of the project plan. Sometimes simply identifying risks can help avert them, but, if that cannot be done, at least you'll have a backup plan available. One caution: if you are dealing with very analytical people, they may go into *analysis*

Identify project risks and develop contingencies to deal with them if they occur.

paralysis here. You are not trying to identify *every possible risk—* just those that are fairly likely.

Begin by looking at the *purpose* of doing whatever is to be done. Develop a problem statement. All actions in an organization should be taken to achieve a result, which is another way of saying "solve a problem." Be careful here to identify what the end user really needs to solve the problem. Sometimes we see projects in which the team thinks a solution is right for the client but that solution is never used, resulting in significant waste to the organization.

> Consider the little mouse, how sagacious an animal it is which never entrusts its life to one hole only.
>
> —Plautus (254–184 B.C.)

Use the Work Breakdown Structure (discussed in chapter 4) to divide the work into smaller chunks for which you can develop accurate estimates for duration, cost, and resource requirements.

Project Planning Steps

The basic planning steps are as follows. Note that some of these topics are covered in the next chapter.

> Be sure the project really satisfies the customer's needs, rather than being what the team thinks the customer should get!

- ▶ Define the problem to be solved by the project.
- Develop a mission statement, followed by statements of major objectives.
- Develop a project strategy that will meet all project objectives.
- Write a scope statement to define project boundaries (what *will* and *will not* be done).
- ▶ Develop a Work Breakdown Structure (WBS).
- Using the WBS, estimate activity durations, resource requirements, and costs (as appropriate for your environment).

➥ Prepare the project master schedule and budget.

▶ Decide on the project organization structure — whether matrix or hierarchical (if you are free to choose).

➥ Set up the project notebook.

■ Get the plan signed off by all project stakeholders.

■■
Key Points to Remember

■ If you have no plan, you have no control.

➥ The people who must execute a plan should participate in preparing it.

➥ Have the plan signed off in a meeting, not by sending it through the interoffice mail.

➥ Keep all project documentation in a project notebook.

➥ Use exit criteria to determine when a milestone has actually been achieved.

➥ Require that changes to the project plan be approved before making them.

Risk management should be part of all project planning.

A paradigm is a belief about what the world is like.

■ Planning is answering the "who, what, when, and how" questions.

Logistics refers to supplying people with materials and supplies they need to do their jobs.
■■

Exercise . . . .

We have talked about strategy, tactics, and logistics. Which must be decided first? What is the function of tactics? When would you plan for logistics?

Developing a Mission, Vision, Goals, and Objectives for the Project

Before a project team does any work, it should spend time ensuring that it has a *shared understanding* of where it is going. The terms used to define that destination are mission, vision, goals, and objectives. And it is at this very early stage that projects tend to fail, because everyone takes for granted that "we all know what the mission is."

Defining the Problem

Every project solves a problem of some kind, but people are inclined to skip over the definition of the problem. This is a big mistake. The way you define a problem determines how you will solve it, so it is critical that a proper definition be developed. For example, too often a problem is defined in terms of a solution. **A** person may say, "I have a problem. My car has quit, and I have no way to get to work. How am I going to get my car repaired, because I have no money to do it?"

The problem has essentially been defined as "How do I

repair my car?" The actual problem, however, at its most funda-
mental level, is that the person has no way to get to work--or so
he says. But could he ride the bus, go with a coworker, or ride a
bike until he has the money to have the car repaired? It is true
that having no money to repair the car is a problem, but it is
important to distinguish between the basic or core problem and
those at another level.

I once heard a sales manager berate a salesman, saying,
"The company has spent a lot of money developing this new
product, and none of you are selling it. If you don't get out
there and sell this product, I'm going to find myself some sales-
people who can sell!"

It is clear how he has defined the problem—he has a group of
salespeople who can't sell. However, given that none of them can
sell the product, I am sure he is wrong.
There is something wrong with the prod-
uct or the market, or the competition is
killing them. You are very unlikely to have
all bad salespeople!

Nevertheless, this manager has defined
the problem in terms of people, and that is
the way it must be solved. Imagine that he
replaces all of the salespeople. He **will** still
have the same problem, because he has not
addressed the actual cause.

People sometimes define a problem as
a goal. A goal in itself is not a problem. It is
when there are obstacles that make it diffi-
cult to reach the goal that one has a problem. Given this defini-
tion of a problem, we can say that problem solving involves find-
ing ways to deal with obstacles: they must be overcome,
bypassed, or removed.

> A problem is a gap
> between where you
> are and where you
> want to be, with
> obstacles existing
> that prevent easy
> movement to close
> the gap.

Confusion of Terms

Suppose a person tells you that she is taking a new job in a distant
city, and she plans to move there. She immediately realizes that

she must find a place to live. So she says, "I have a problem. I have to find a place to live."

You ask her what her mission is. "To find a place to live," she says.

And how about her vision? "To have a place to live," she answers, a little confused.

No wonder she is confused. *All* three statements sound alike! She needs to understand the difference between them if she is to solve this problem.

Remember, a problem is a gap. Suppose we were to ask her to tell us where she wants to be when her problem is solved. She would say, "I would have a place to live in the new city."

"And where are you now?" you ask.

"I have no place to live," she says.

Then the gap is between having a place and not having one. This can be stated simply as "I have no place to live." And, indeed, this is the problem she is trying to solve.

But—would just anyplace be okay? Of course not. She doesn't want to live under a bridge, although homeless people sometimes do. So if you ask her, "What kind of place are you looking for?" she can tell you.

"It needs to have three bedrooms, the house must be of a certain size, and I prefer a certain style," she says. This is her vision for the kind of place she wants to live in. That vision literally paints a picture in her mind, and, when she finds a place that comes close to that picture, she **will** have "arrived" at her destination. This is the function of vision—it defines "done."

Her mission, then, is to find a place that conforms to her vision. Another way to say this is that the mission of a project is always to achieve the vision. In doing so, it solves the stated problem. So you may want to diagram it as shown in Figure 3-1. Note that the vision has been spelled out as a list of things she must have, along with some that she wants to have and a few that would be nice to have if she could get them.

Figure 3-1. Chevron showing mission, vision, and problem statement.

Problem: I have no place to live.		
MUSTS	**WANTS**	**NICE**
3 bedrooms 2,500 sq. ft. 2-cargarage I-acre lot large family room	room for homeoffice basement	fireplace in family room

Mission:
To find a place that meets all musts and as many of the others as possible.
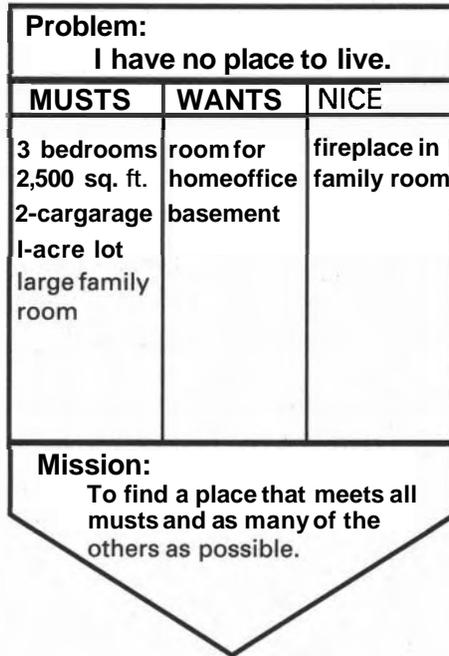
The Real World

Okay, now we know the difference between the mission, vision, and problem, but in the "real world" you never get them in this order. Your boss or project sponsor **will** say, "Here is your mission," without any mention of a problem statement. It is possible that some discussion of the sponsor's vision of the end result **will** take place, but even that may be fairly sketchy. So the first order of business for a project team is to develop these into a form that everyone will accept.

The major "political" problem you may encounter is that the sponsor **will** undoubtedly have given you a mission based on his definition of the problem to be solved. Sometimes his definition **will** be incorrect, and you **will** have to confront this. Otherwise, you will spend a lot of the organization's money, only to find that you have developed the right solution to the wrong problem.

The Real Mission of Every Project

I said earlier that the mission is always to achieve the vision. However, I should add that the vision you are trying to achieve is the one the customer holds. Another way to say this is that you are trying to satisfy the customer's needs. That is the primary objective. Your motive may be to make a profit in the process, but the mission is *always* to meet the needs of the customer. That means, of course, that you must know what those needs are, and sometimes this isn't easy, because even the customer isn't clear about them. So you have to translate or interpret as best you can. Your best safeguard is to keep the customer involved in the project from concept to completion so that there is a constant check on whether what you are doing **will** achieve the desired result.

The mission of the project can be written by answering two questions:

1. What are we going to do?

2. For whom are we going to do it?

In the previous edition of this book, I suggested that you also state how you **will** go about meeting those customer needs, but I have since decided that this should not be part of the mission statement itself. The mission statement defines "what" you are doing; "how" you are going to do it is project strategy and should be dealt with separately.

Some Examples

When AMACOM called me to update this book, it defined the problem this way: the first edition was considered a bit dated, and it did not contain information on Project Management Institute and the Project Management Body of Knowledge (PMBOK®) (which was presented in chapter 1 of this edition). I also felt, on the basis of the experience I have had since the first edition was released, that some sections of the book could be clarified for the reader.

Problem Statement

If I were to write a problem statement for this project, then, it would say:

> The first edition of *Fundamentals of Project Management* lacks information on the PMBOK and is seen as being out-of-date by readers. In addition, some material should be clarified.

Vision

The vision for the book remains about the same as for the first edition. The original series was designed to provide readers with a quick, easy read on the subject covered. The books are kept to under 200 pages and are written in down-to-earth language. All are published in paperback. I also add to this my own component of the vision: I like my books to be well illustrated, to enhance the reader's understanding of the material.

Mission Statement

The mission for this project is to develop a new edition of the book that includes PMBOK material and that clarifies some material that was unclear in the first edition. The target audience (who it is for) is senior managers and other readers who need a quick overview of project management without going into a great deal of depth.

Boeing's 777 Airplane

In his book *21st Century Jet,* Karl Sabbagh (1996) explains that Boeing's customers started asking for a plane smaller than the 747. Boeing had just finished developing the 767, so it suggested that customers buy it. "It's too small," said the airlines.

Other customers asked for a plane a little larger than the 767. Boeing suggested the 747. "It's too big," said the airlines. It took Boeing nearly two years to realize that the airlines wanted a

plane intermediate in size between the 747 and the 767. When it finally got the message, it developed the 777, which is the subject of Sabbagh's book.

A problem statement for this project might have said:

We have no plane intermediate in size between the 747 and the 767 that can carry passengers over a transpacific route.

(Please note that this is my statement. Boeing did not provide it to me, and I don't know if it would fully agree with it.)

Developing Project Objectives

Once a mission statement has been developed, you can write your project objectives. Note that objectives are much more specific than the mission statement itself and define results that must be achieved in order for the overall mission to be accomplished. Also, an objective defines the desired *end result.*

I may want to finish this chapter by 10 o'clock this morning. That is my desired outcome or result—my *objective.* The way in which I achieve that objective is to perform a number of tasks. These might include typing text into my computer, reviewing some other literature on the topic about which I am writing, calling a colleague to ask a question for clarification, and printing out the chapter, proofing it, and entering some revisions into my computer.

> **Goal setting has traditionally been based on past performance. This practice has tended to perpetuate the sins of the past.**
> —J. M. Juran

The following acronym may help you remember the essential qualities that a statement of objectives must have. We say that an objective must be SMART, each letter standing for a condition as follows:

Specific

Measurable

Attainable

Realistic

Time-limited

Dr. W. Edwards Deming has raised some serious questions about the advisability of trying to quantify goals and objectives. He argued, in Out of the Crisis (1986), that there is no point in setting quotas for a manufacturing process to reach. If the system is stable, he argued, then there is no need to specify a goal, since you will get whatever the system can produce. A goal beyond the capability of the system can't be achieved.

On the other hand, if the system is not stable (in the statistical sense of the word), then again there is no need to specify a quota, since there is no way to know what the capability of the system is (Deming, p. 76).

In project work, we may know the capability of a person by looking at his or her past performance, but unless you have a large number of samples, you have no way of knowing exactly what the person can do, since there is always variability in people's performance. Furthermore, it does no good to base a quota on what someone else has done. The quota must be valid for the person who is going to do the job this time.

An *objective* specifies a desired end result to be achieved.

A *task* is an activity performed to achieve that result.

An objective is usually a *noun,* whereas a task is a *verb.*

We all know that some people are capable of more output than others. So defining the measurement and attainability aspects of goal or objective setting is very difficult. I go into this more in chapter 4 when I discuss time estimating.

I have found the following two questions to be useful both in setting objectives and in monitoring progress toward those objectives:

What is our desired *outcome?* This is called the outcome *frame* and helps keep you focused on the result you are trying to achieve, rather than on the effort being expended to get there.

► How will we know when we achieve *it?* I call this the evidence question. This question is very useful for establishing exit criteria for objectives that cannot be quantified.

Some Examples of Objectives

► Our objective is to develop a one-minute commercial to solicit contributions to WXYZ to air on local TV stations by June 5, 2002.

► Our objective is to raise $600,000 in funds from local viewers by September 18, 2002.

The Nature of Objectives

Note that these examples of objectives do not say how they will be achieved. I consider an objective to be a statement that tells me what result is to be achieved. The how is problem solving, and I prefer to keep that open so that solutions can be brainstormed later. If the approach is written into the objective statement, it may lock a team into a method that is not really best for the project.

Assessing Project Risks

Once you have established your objectives, you can develop plans for how to achieve them. Unfortunately, the best plans sometimes don't work. One safeguard in managing projects is to think about the risks ahead that could sink the job. This can be done for critical objectives and for other parts of the plan.

The simplest way to conduct a risk analysis is to ask, "What could go wrong?" or "What could keep us from achieving our objective?" It is usually best to list the risks first, then think about contingencies for dealing with them. One way to look at risk is to divide a flip chart page in half, have the group brainstorm the risks, which you write down on the left side of the page, and then go back and list the contingencies— things you can do to manage the risks if they do materialize. An example of a risk analysis for a photography project is shown in Figure 3-2.

It is helpful to assess risks of failure of the following:
The schedule
► The budget
► Project quality
Customer satisfaction

One benefit of doing a risk analysis in this manner is that it may help you avert some risks. When you cannot avert a risk, you **will** at least have a backup plan. Unexpected risks can throw a project into a tailspin.

I mentioned this point previously, but it bears repeating: you are not trying to identify *every* possible risk, just some of the more likely ones. This point should be made to team members who are highly analytical or who perhaps have a tendency to be negative in general. Also, risk analysis always has a

Risk analysis should not lead to analysis paralysis!

Figure 3-2. Risk analysis example.

What could go wrong?	Contingency
1. Exposure wrong	Bracket the exposure
2. Shots unacceptable	Take extra photos
3. Film lost or damaged	Hand carry to client
4. Weather delays	Allow extra time

positive thrust—that is, you are asking, "If it happens, what will we do about it?" You don't want people to play "Ain't it awful!"

■■■

Key Points to Remember

The way a problem is defined determines how you will solve it.

A problem is a gap between where you are and where you want to be, with obstacles making it hard to reach the goal. A goal by itself is not a problem. Obstacles must exist for there to be a problem.

Vision is what the final result will "look like." It defines "done."

▶ The mission is to achieve the vision. It answers the two questions "What are we going to do?" and "For whom are we going to do it?"

Objectives should be SMART.

You can identify risks by asking, "What could go wrong?"

Planning Exercise . . . .

Choose a project that you are going to do or perhaps have just started. Answer the questions that follow to the best of your ability. If you need to confer with others to answer some of them, fine. Remember, the people who have to follow the plan should participate in preparing it.

▶ What are you trying to achieve with the project? What need does it satisfy for your customer? Who exactly is going to actually use the project deliverable(s) when it is finished? (That is, who is your real customer?) What will distinguish your deliverable from those already available to the customer?

▶ Write a problem statement on the basis of your answers to the first question. What is the gap between where you are now and

where you want to be? What obstacles prevent easy movement to close the gap?

▶ Write a mission statement, answering the two basic questions:

What are we going to do?
For whom are we going to do it?

▶ Talk to your customer about these issues. Do not present your written statements to her. Instead, see whether you can get confirmation by asking open-ended questions. If you can't, you may have to revise what you have written.

Using the Work Breakdown Structure to Plan a Project

n a previous chapter, I said that planning answers the questions "What must be done?," "How long will it take?," and "How much will it cost?" Planning the *what* is vital; projects frequently fail because a significant part of the work is *forgotten.* In addition, once tasks have been identified, the time and resource requirements must be determined. This is called *estimating.*

A major problem in project planning is determining how long tasks will take and what it will cost to do them. Inaccurate estimates are a leading cause of project failures, and missed cost targets are a common cause of stress and recrimination in project management.

The most useful tool for accomplishing all of these tasks is the Work Breakdown Structure (WBS). The idea behind the WBS is simple: you can subdivide a complicated task into smaller tasks, until you reach a level that cannot be further subdivided. At that point, it is usually easier to estimate how long the small task will

take and how much it will cost to perform than it would have
been to estimate these factors for the higher levels.

Nevertheless, it is still not easy to estimate task durations for
activities that have never been performed before. Because this is
the typical situation in engineering hardware and software devel-
opment projects, we might expect many of these estimates to be
in error, and this seems to be demonstrated by experience. Still,
the Work Breakdown Structure makes it easier to estimate
knowledge tasks than any other tool we have.

A Simple Example

As an example, if I want to clean a room (see Figure 4-I), I might
begin by picking up clothes, toys, and other things that have been
dropped on the floor. I could use a vacuum cleaner to get dirt out
of the carpet. I might wash the windows and wipe down the
walls, then dust the furniture. All of these activities are *subtasks*
performed to clean the room.

As for vacuuming the room, I might have to get the vacuum
cleaner out of the closet, connect the hose, plug it in, push the
vacuum cleaner around the room, empty the bag, and put the
machine back in the closet. These are still smaller tasks to be

Figure 4-1. WBS diagram to clean a room.
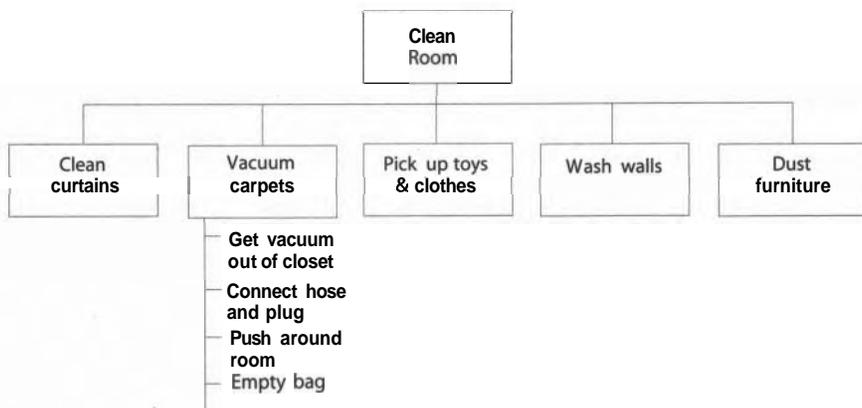
performed in accomplishing the subtask called *vacuuming.* The diagram in Figure 4-1 shows how this might be portrayed in WBS format.

Note that we do not worry about the *sequence* in which work is performed when we do a WBS. That **will** be worked out when we develop a schedule. However, you **will** probably find yourself thinking sequentially, as it seems to be human nature to do so. The main idea of doing a WBS is to capture all of the tasks. So if you find yourself and other members of your team thinking sequentially, don't be too concerned, but don't get hung up on trying to diagram the sequence or you will slow down the process of task identification.

> A Work **Breakdown Structure** *does not* show *the sequence* **in which work is performed!** Such sequencing is **determined when** a schedule is **developed.**

The typical WBS has three to six levels, and these can be named as shown in Figure 4-2. It is, of course, possible to have projects that require a lot more levels. Twenty levels is considered to be the upper limit, and that is a huge project. Note that level 1 is called the *program* level. The difference between a program and a project is just one of degree.

An example of a program is the development of an airplane. For example, the WBS for Boeing's 777 airplane program might have been drawn as shown in Figure **4-3.** Notice that the engine, wing, and avionics are large enough jobs to be called projects in their own right. In fact, the program manager's job is to make sure that the projects are all properly integrated. The engine mounts on the wing, so, somewhere in the structure to develop the engine, there will be an activity called "Design wing mounts." And for the wing, there will be an activity called "Design engine mounts." If these are not coordinated properly, you will wind up with an engine that won't mount on the wing. The job of coordinating these is called *system integration.*

Figure 4-2. WBS level names.

1. **Program**

2. **Project**

3. **Task**

4. **Subt ask**

5. **Work Package**

6. **Level of Effort**
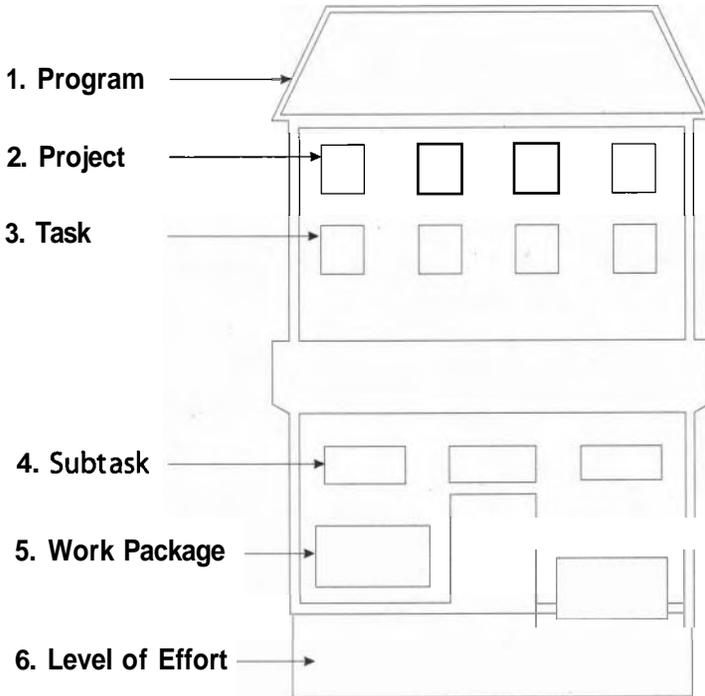
Guidelines for Developing the WB S

One important question in constructing a WBS is "When do you stop breaking down the work?" The general guideline is that you stop when you reach a point where either you can estimate time and cost to the desired degree of accuracy or the work will take an amount of time equal to the smallest units you want to schedule. If, for instance, you want to schedule to the nearest day, you break down the work to the point where tasks take about a day to perform. If you are going to schedule to the nearest hour, then you stop when task durations are in that range.

Stop breaking down work when you reach a low enough level t o do an estimate of the desired accuracy.

Figure 4-3. Partial WBS for the 777 development program.
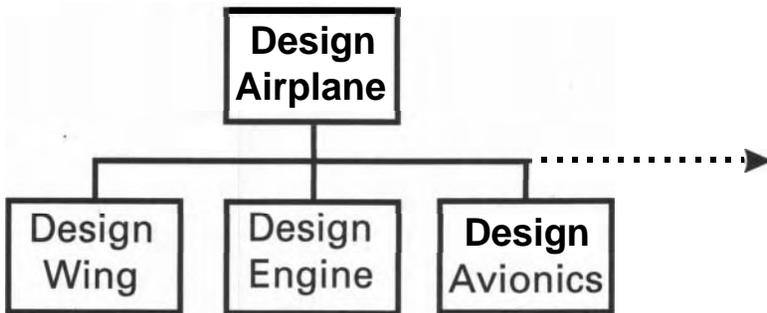
Remember the rule that the people who must do the work should participate in planning it? That applies here. Usually a core group identifies top-level parts of the WBS; those parts are further refined by other members of the team and then integrated to obtain the entire WBS.

One important point: the WBS should be developed before the schedule. In fact, the WBS is the device that ties the entire project together. It allows resources to be assigned and estimates of time and cost to be made and shows the scope of the job in graphic form. Later, as the project is tracked, the work can be identified as falling in a particular box in the WBS.

There is at least one software package, SuperProject Expert™, that prints a WBS after schedule data have been entered. That is a nice feature, since it gives a graphically attractive WBS, but the rough drawing should be made before you use the scheduling software. The reason is quite simple: until everyone has agreed that all tasks have been identified, it is misleading to develop a schedule. You cannot be sure that the critical path identified by a partial schedule **will** be the same for the full schedule.

> The WBS should always be devel-
>
> oped **before the schedule is worked**
>
> **out, but without trying to identify**
>
> **the sequence of**
>
> **activities.**

There are a number of approaches to developing the WBS. Ideally, you proceed top-down, following development of a good problem statement and mission statement. As I have mentioned, however, the mind does not always operate in such nice, linear fashion; as you develop the WBS, you may sometimes find that it helps you to understand the job better. For that reason, I am not a purist about doing things in a specific order. You do what works best for you.

> **A WBS does not have to be symmetrical. All paths do not have to go down to the same level.**

The WBS does not have to be symmetrical. That is, all paths need not be broken down to level 6 (or whatever level you stop at). Since the rule is to break work down to a level sufficient to achieve the estimating accuracy you desire, one path may take six levels, while another may need only three.

Uses of the WBS

As I have said, the WBS is a good way to show the scope of a job. If you have ever given someone an estimate for project cost or time and seen the person's horrified look, you know that they are seeing the project in their mind as much simpler than it is. When you show a project in WBS form, it is clear to most individuals why the job costs so much. In fact, I have had the experience of the planning group members

> **The WBS is a good way to portray the scope of a project.**

themselves being overwhelmed by the complexity and magnitude of the WBS. If it impresses them, think of the impact on the outsider.

Assigning responsibility for tasks is another important use of the WBS. Each task to be performed should be assigned to a particular person who will be responsible for its completion. These assignments can then be listed on a separate form, often called a Responsibility Chart (see Figure 4-4).

Figure 4-4. Responsibility chart.

Linear Responsibility Chart															
Project:			Date Issued:					Sheet Number:			of				
Manager:			Date Revised:					Revision No.			File: LRCFORM.61				
				Project Contributors											
Task Descriptions															
CODES: 1 = ACTUAL RESPONSIBILITY; 2 = SUPPORT; 3 = MUST BE NOTIFIED; BLANK = NOT INVOLVED															

Estimating Time, Costs, and Resources

Once the work is broken down, you can estimate how long it **will** take. But how? Suppose I ask you how long it **will** take to sort a standard deck of playing cards that has been thoroughly shuffled into numerical order by suit. How would you answer that question?

The most obvious way would be to try the task several times and get a feeling for it. But if you didn't have a deck of cards handy, you would probably think about it, *imagine* how long it would take, and give me an answer. People generally give me answers ranging from two minutes to ten minutes. Mv tests indicate that about three minutes is average for most adults.

An estimate can be made only by starting with the assumption that a certain resource will be assigned.

Suppose, however, we gave the cards to a child about four or five years old. It might take a lot longer, since the child would not be that familiar with the

sequence in which cards are ordered, and perhaps not yet even that comfortable with counting. So we must reach a very important conclusion: you cannot do a time or cost estimate without considering who will actually perform the task. Second, you must base the estimate on historical data or a mental model. Historical data are best.

> **Parkinson's Law: Work expands to fill the time allowed.**

Generally we use average times to plan projects. That is, if it takes three minutes on average for adults to sort a deck of cards, I would use three minutes as my estimate of how long it will take during execution of my project. Naturally, when I use averages, in reality some tasks **will** take longer than the time allowed and some should take less. Overall, however, they should average out.

That is the idea, anyway. Parkinson's Law discredits this notion, however. Parkinson said that work always expands to fill the time allowed. That means that tasks may take longer than the estimated time, but they almost never take less. One reason is that when people find themselves with some time left, they tend to refine what they have done. Another is that people fear that if they turn work in early, they may be expected to do the task faster the next time or that they may be given more work to do.

> **We must be careful not to penalize workers who perform better than expected by loading them down with excessive work,**

This is a very important point: if people are penalized for performing better than the target, they will quit doing so. We also have to understand variation. If the same person sorts a deck of cards over and over, we know the sort times will vary. Sometimes it will take two minutes, while other times it will take four. The average may be three, but we may expect that half the time it **will** take three minutes or less and half the time it **will** take three minutes or more. Very seldom will it take *exactly* three minutes.

> **An exact estimate is an oxymoron!**

The same is true for allproject tasks. The time it takes to perform them will vary, because of forces outside the person's control. The cards are shuffled differently every time. The person's attention is diverted by a loud noise outside. He drops a card while sorting. He gets tired. And so on.

Can you get rid of the variation? No way.

Can you reduce it? Yes—through practice, by changing the process by which the work is done, and so on. But it is important to note that the variation will always be there, and we must recognize and accept it.

The Hazards of Estimating

Consider the case of Karen. One day her boss stopped by her desk at about one o'clock. "Need for you to do an estimate for me," he told her. "Promised the Big Guy I'd have it for him by four o'clock. You with me?"

Karen nodded and gave him a thin smile. The boss described the job for her. "Just need a ballpark number," he assured her and drifted off.

Given so little time, Karen could compare the project her boss described only to one she had done about a year before. She added a little for this and took a little off for that, put in some contingency to cover her lack of information, and gave the estimate to the boss. After that, she forgot all about the job.

Two months passed. Then the bomb was dropped. Her boss appeared, all smiles. "Remember that estimate you did for me on the xyz job?"

She had to think hard to remember, but, as her boss droned on, it came back to her. He piled a big stack of specifications on her desk. "It's your job now," he told her and drifted off again into manager dreamland.

As she studied the pile of paper, Karen felt herself growing more concerned. There were significant differences between this set of specs and what her boss had told her when she did the estimate. "Oh well, I'm sure he knows that," she told herself.

Finally she managed to work up a new estimate for the job on

the basis of the real specs. It was almost 50 percent higher than the ballpark figure. She checked her figures carefully, assured herself that they were correct, and went to see her boss.

He took one look at the numbers and went ballistic. "What are you trying to do to me?" he yelled. "I already told the old man we would do it for the original figure. I can't tell him it's this much more. He'll kill me."

"But you told me it was just a ballpark number you needed," Karen argued. "That's what I gave you. But this is nothing like the job I quoted. It's a lot bigger."

"I can't help that," her boss argued. "I already gave him the figures. You'll have to find a way to do it for the original bid."

Naturally, you know the rest of the story. The job cost even more than Karen's new estimate. There was a lot of moaning and groaning, but, in the end, Karen survived. Oh, they did send her off to a course on project management—hoping, no doubt, that she would learn how to estimate better in the future.

> One of the primary causes of project failures is that ballpark estimates become targets.

Guidelines for documenting estimates:

■ Show the percent tolerance that is likely to apply.

▶ Tell how the estimate was made and what assumptions were used.

▶ Specify any factors that might affect the validity of the estimate (such as time—will the estimate still be valid in six months?).

Could you fault Karen for anything? Well, perhaps. If she failed to tell the boss that a ballpark estimate may have a tolerance of perhaps ±25 percent but that the margin of error can range from –1 0 percent to +100 percent, then she allowed him to think that the estimate was better than it was. Also, she should have documented all working assumptions, explaining how she

did the estimate, what project she had used for comparison, and so on. Then, if management still pulled a whammy on her, at least she would have had some protection. In fact, it is impossible to make sense of any estimate unless these steps are taken, so they should be standard practice.

Consensual Estimating

In recent years, a new method of estimating knowledge work has been developed that seems to work better than older techniques. Rather than have individuals estimate task durations, the new method asks at least three people to estimate each activity in the project that they know something about. They do this without discussing their ideas with one another. They then meet to find out what they have put on paper. In a typical situation, there may be a range of times, such as, for example, ten days, twelve days, and thirty days, in which two of the estimates are close together, but one is very different. How do you handle the discrepancy?

The best approach is to discuss what each person was considering when he or she made the estimate. It may be that the person who put down thirty days was thinking about something that the other two had overlooked. Or, conversely, they may convince the thirty-day person that his number is way too high and get him to come down to a figure nearer their estimates. In any case, they try to arrive at a number that they all can support. This is called consensus.

There are three advantages to this approach. First, no one person is on the hook for the final number. Second, inexperienced people learn to estimate from those more experienced. Third, they are likely to collectively consider more issues than any one person would do working alone. For that reason, you are more likely to get an accurate estimate, although it is important to remember that it is still not exact!

Improving Estimating Ability

People cannot learn unless they receive feedback on their performance. If you went out every day and ran 100 yards, trying to

improve your speed, but you never timed yourself, you would have no idea whether you were getting better or worse. You could be doing something that slowed you down, but you wouldn't know it. In the same way, if you estimate task durations but never record the actual time it takes to do the task, you are never going to get better at estimating. Furthermore, you have to track progress by recording times daily. If you record times once a week, I can promise you that you will be just guessing, and that won't be helpful.

Key Points to Remember

■ Do not try to work out sequencing of activities when you develop a WBS. You will do that when you develop a schedule.

■ A WBS ties the entire project together. It portrays scope graphically, allows resources to be assigned, permits estimates of time and costs to be developed, and thus provides the basis for the schedule and the budget.

❞ An estimate is a guess, and an exact estimate is an oxymoron!

❞ Be careful that ballpark estimates don't become targets.

❞ Consensual estimating is a good way to deal with activities for which no history exists.

❞ No learning takes place without feedback. Estimate; then track your actual time if you want to improve your estimating ability.

Exercise . . . .

Following is a list of tasks to be performed in preparation for a camping trip. Draw a WBS that places the tasks in their proper rela-

tionship to one another. The solution is contained in the Answers section.

Arrange for supplies and equipment.

Select campsite.

Make site preparations.

Make site reservation.

Arrange time off from work.

Select route to site.

Prepare menu for meals.

Identify source of supplies and equipment.

Load car.

Pack suitcases.

Purchase supplies.

Arrange camping trip (project).

Scheduling Project Work

One of the primary features that distinguishes project management from general management is the special attention to scheduling. Remember from chapter 1 that Dr. J. M. Juran says a project is a problem scheduled for solution.

Unfortunately, some people think that project management is nothing *but* scheduling, and this is incorrect. Scheduling is just one of the tools used to manage jobs and should not be considered the primary one.

Project management is not just scheduling.

People today tend to acquire scheduling software, of which there is an abundance, and think that will make them instant project managers. They soon find that that idea is wrong. In fact, it is nearly impossible to use the software effectively unless you understand project management (and scheduling methodology in particular).

I do have one suggestion about soft-

Suggestion: Whatever scheduling software you choose, get some professional training on how to use it.

58

ware. Whatever you pick, get some professional training in how to use it. In the early days of personal computers, there was a pretty significant difference between the low-end and the high-end software that was available. The low-end packages were pretty easy to use, whereas the high-end ones were not.

The gap between low- and high-end software has closed to the point that this is no longer true. They are all difficult to use now, and the training materials (tutorials and manuals) that come with the software are often not very good. In addition, it is hard to find time to work through a tutorial without being interrupted several times, which means that self-learning is difficult. The most efficient way is to take a class.

Do check out the instructor's knowledge of project management. Some of the people teaching the software know very little about project management itself, and, when you have questions, they can't answer them.

You should expect to spend from two to three days of classroom time becoming really proficient with the software. That is still a good investment, considering the time the software can save you in the long run.

A Brief History of Scheduling

Until around 1958, the only tool for scheduling projects was the bar chart (see Figure 5-1). Because Henry Gantt developed a complete notational system for showing progress with bar charts, they are often called Gantt charts. They are simple to construct and read and remain the best tool to use for communicating to team members what they need to do within given time frames. Arrow diagrams tend to be too complicated for some teams. Nevertheless, it is often helpful to show an arrow diagram to the people doing the work so that they understand interdependencies and why it is important that they complete certain tasks on time.

Bar charts do have one serious drawback—it is very difficult to determine the impact of a slip on one task on the rest of the

Figure 5-1. Bar chart.
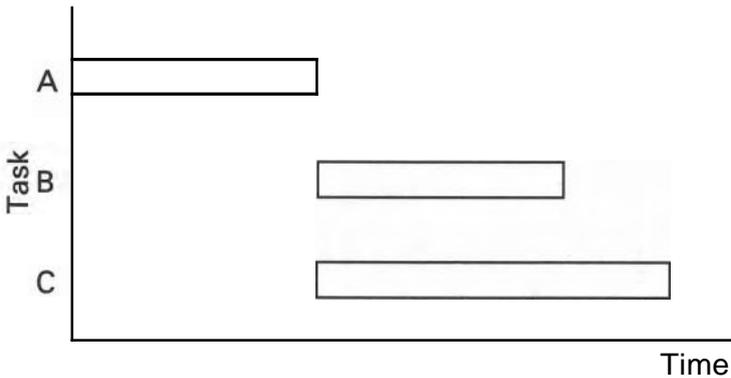
project (e.g., if Task A in Figure 5-1 gets behind, it is hard to tell how this will affect the rest of the work). The reason is that the bar chart (in its original format) did not show the interdependencies of the work. (Contemporary software does show links between bars, making them easier to read. The actual name for these bar charts is "time-line critical path schedules.")

To overcome this problem, two methods of scheduling were developed in the late 1950s and early 1960s, both of which use arrow diagrams to capture the sequential and parallel relationships among project activities. One of these methods, developed by Du Pont, is called *Critical Path Method (CPM),* and the other, developed by the U.S. Navy and the Booze, Allen, and Hamilton consulting group, is called *Program Evaluation and Review Technique (PERT).* Although it has become customary to call all arrow diagrams PERT networks, strictly speaking the PERT method makes use of probability techniques, whereas CPM does not. In other words, with PERT it is possible to calculate the probability that an activity will be completed by a certain time, whereas that is not possible with CPM.

CPM: Critical Path Method

PERT: Program Evaluation and Review Technique

Network Diagrams

To show the sequence in which work is performed, diagrams like those in Figure 5-2 are used. In these diagrams Task A is done before B, while Task C is done in parallel with them.

The network in the bottom half of Figure 5-2 uses *activity-on-arrow* notation, in which the arrow represents the work being done and the circle represents an event. An event is *binary;* that is, it has either occurred or it has not. An activity, on the other hand, can be partially complete. Note that this is a special use of the word *event.* We speak of a football game as an event, even though it spans time. In scheduling terminology, however, an event is a specific point in time where something has just started or has just been finished.

Figure 5-2. Arrow diagrams.

An activity-on-node network

Activity A → Activity B → Activity D
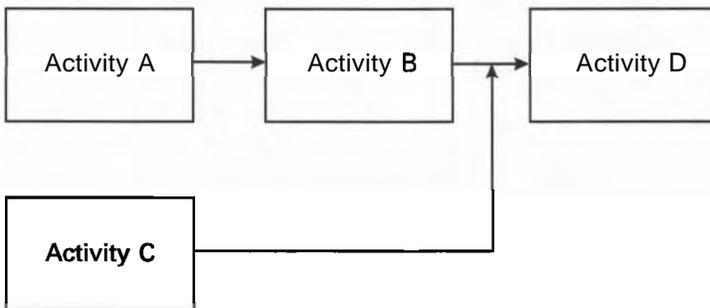
Activity C

An activity-on-arrow network

Activity A 2 Activity B
1 Activity C 3 Activity D 4
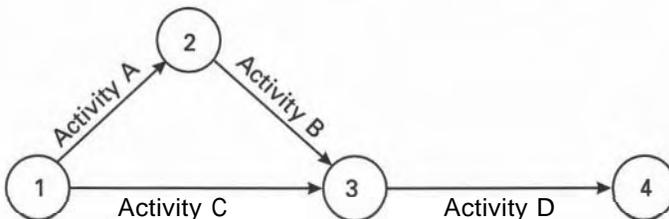
The network in the top half of Figure 5-2 uses activity-on-node notation, which shows the work as a box or node, and the arrows show the sequence in which the work is performed. Events are not shown in activity-on-node networks unless they are *milestones*—points in the project at which major portions of the work are completed.

Why two forms of diagrams? Probably a tyranny to confuse the uninitiated. Actually, it simply happens that the schemes were developed by different practitioners.

Is one better than the other? No. They both get the same results in figuring out when work is supposed to be completed. Both forms are still used, although activity-on-node is used a bit more than the other, simply because much of today's personal computer software is programmed to use node notation.

What is the benefit of using either CPM or PERT? The main advantage is that you can tell whether it is possible to meet an important project completion date, and you can also tell exactly when various tasks must be finished in order to meet that deadline. Furthermore, you can tell which tasks have some leeway and which do not. In fact, both CPM and PERT determine the critical path, which is defined as the longest series of activities (that can't be done in parallel) and which therefore governs how early the project can be completed.

The *critical* path is the longest path through a project network. Because it has no slack, all activities on the critical path must be completed as scheduled, or the end date will begin to slip—one day for each day a critical activity is delayed.

The Reason for Scheduling

Naturally, the primary reason for scheduling a project is to ensure that the deadline can be met. Most projects have a deadline imposed. Furthermore, since the critical path method helps iden-

tify which activities will determine the end date, it also helps guide how the project should be managed.

However, it is easy to get carried away with scheduling and spend all of your time updating, revising, and so on. The scheduling software in use today should be viewed as a tool, and managers should not become slaves to the tool.

It is also very easy to create schedules that look good on paper but don't work in practice. The main reason is usually that resources are not available to do the work when it comes due. In fact, unless resource allocation is handled properly, schedules are next to useless. Fortunately, today's scheduling software handles resource allocation fairly well, but we leave discussion of the methods used to the software manuals. In this book, we simply examine how networks are used to show us where we need to manage.

One company found that when it stopped having people work on multiple projects, workers' productivity doubled!

I am often-told that scope and priorities change so often in a given organization that it doesn't make sense to spend time finding critical paths. There are two points worth considering here. One is that if scope is changing often in a project, not enough time is being spent doing up-front definition and planning. Scope changes most often occur because something is forgotten. Better attention to what is being done in the beginning usually reduces scope creep.

Second, if priorities are changing often, management does not have its act together. Generally, the organization is trying to tackle too much work for the number of resources available. We all have "wish lists" of things we want to do personally, but we have to put some of them on hold until time and/or money become available. The same is true of organizations. Experience shows that when you have individuals working on many projects, productivity suffers. One company found, as an example, that when it stopped having people work on multiple projects,

employees' productivity ***doubled!*** That obviously is highly significant.

What does CPM have to do with this? Knowing where the critical path is in a project allows you to determine the impact on the project of a scope or priority change. You know which activities **will** be impacted most heavily and what might need to be done to regain lost time. In addition, managers can make informed decisions when you can tell them the impact of changes to the project. Thus, CPM can be an invaluable tool, when used properly.

Definitions of Network Terms

ACTIVITY	An activity always consumes time and may also consume resources. Examples include paperwork, labor, negotiations, machinery operations, and lead times for purchased parts or equipment.
CRITICAL	A critical activity or event is one that must be achieved by a certain time, having no latitude (slack or float) whatsoever.
CRITICAL PAM	The critical path is the longest path through a network and determines the earliest completion of project work.
EVENTS	Beginning and ending points of activities are known as events. **An** event is a specific point in time. Events are commonly denoted graphically by a circle and may carry identity nomenclature (e.g., words, numbers, alphanumeric codes).
MILESTONE	This is an event that represents a point in a project of special significance. Usually it is the completion of a major phase of the work. Project reviews are often conducted at milestones.
NETWORK	Networks are called arrow diagrams. They provide a graphical representation of a project plan showing the relationships of the activities.

Constructing an Arrow Diagram

As was pointed out in chapter 4, a Work Breakdown Structure (WBS) should be developed before work on the project is scheduled. Also, we saw that a WBS can contain from two to twenty levels. To illustrate how a schedule is constructed from a WBS, we consider a simple job of maintaining the yard around a home. The WBS is shown in Figure 5-3.

In the case of this WBS, it is appropriate to schedule the tasks at the lowest level. However, this is not always true. Sometimes work **will** be broken down to level 6 but only those activities up to level 5 **will** be entered into the schedule. The reason is that you may not be able to keep level 6 tasks on schedule. That is, you can't manage that tightly. So you schedule at a level that you can manage. This follows the

Don't schedule in

more detail than
you can manage.

general rule that you should never plan (or schedule) in more detail than you can manage. Some projects, such as overhauling a large power generator, are scheduled in increments of hours. Others are

Figure 5-3. WBS to do yard project.
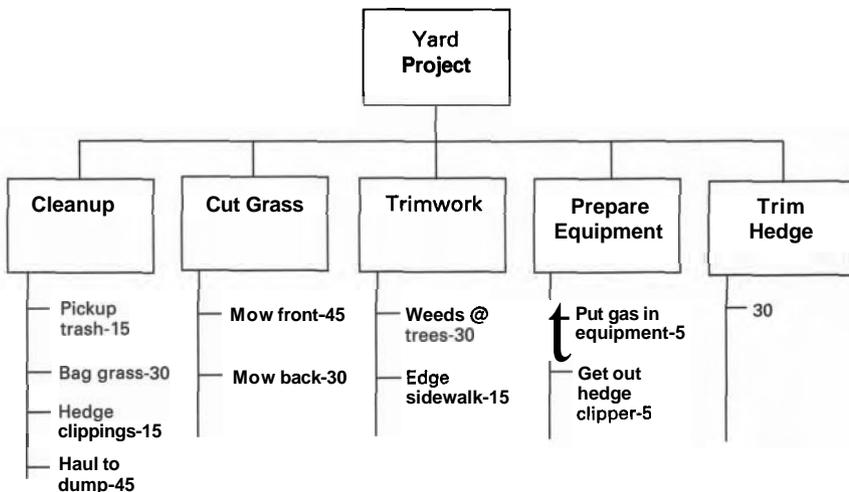
scheduled in days, while some big construction jobs are scheduled to the nearest month.

While planning in too much detail is undesirable, if you plan in too little detail, you might as well not bother. As a practical example, a manager told me that his staff wanted to create schedules showing tasks with twenty-six-week durations. He protested that the staff would never complete such schedules on time. They would *back-end load them,* he argued.

What he meant was that there is a lot of security in a twenty-six-week task. When the start date comes, if the person doing the task is busy, she might say, "I can always make up a day on a twenty-six-week activity. I'll get started tomorrow." This continues until she realizes she has delayed too long. Then there is a big flurry of activity as she tries to finish on time. All the work has been pushed out to the end of the twenty-six-week time frame.

A good rule of thumb to follow is that no task should have a duration much greater than four to six weeks. A twenty-six-week task can probably be broken down into five or six subtasks. Such a plan will generally keep people from back-end loading.

> A good rule of thumb to follow is that no task should have a duration much greater than four to six weeks. For knowledge work, durations should be in the range of one to three weeks, because knowledge work is harder to track than tangible work.

There are two ways you can develop a schedule. One is to begin at the end and work backward until you arrive at the beginning. The second method is to start at the beginning and work toward the end. Usually, it is easiest to start at the beginning.

The first step is to decide what can be done first. Sometimes several tasks can start at the same time. In that case, you simply draw them side-by-side and start working from there. Note the pro-

Figure 5-4. CPM diagram for yard project.
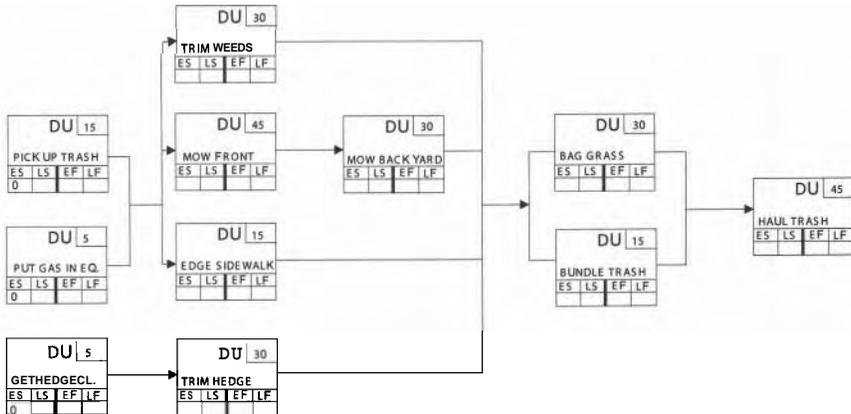
gression in the diagram in Figure 5-4. It sometimes takes several iterations before the sequencing can be worked out completely.

This small project might be thought of as having three phases: *preparation, execution,* and *cleanup.* There are three preparation tasks: *pick up trash, put gas in equipment,* and *get out hedge clipper.* The cleanup tasks include *bagging grass, bundling clippings,* and *hauling trash* to the dump.

In doing this schedule diagram, I have followed a rule of scheduling, which is to *diagram what is logically possible, then deal with resource limitations.* For a yard project, if I have no one helping me, then there really can be no parallel paths. On the other hand, if I can enlist help from the family or neighborhood youth, then parallel paths are possible, so this rule says

Schedules should be developed according to what is logically possible, and resource allocation should be done later.

This will yield the optimum schedule.

go ahead and schedule as *if* it were possible to get help. This is especially important to remember in a work setting, or you **will** never get a schedule put together. You **will** be worrying about who will be available to do the work and end up in analysis paralysis.

Another rule is to keep all times in the same increments. Don't mix hours and minutes—schedule everything in minutes, then convert to hours and minutes as a last step. For this schedule, I have simply kept everything in minutes.

I suggest that you draw your network on paper and check it for logical consistency before entering anything into a computer scheduling program. If the network has logical errors, the computer will just give you a garbage-in, garbage-out result, but it will *look* impressive, having come from a computer.

> **Another rule is to keep all times in the same increments.**

It is also important to remember that there is usually no single solution to a network problem. That is, someone else might draw the arrow diagram a bit differently than you have done. There may be parts of the diagram that have to be done in a certain order, but often there is flexibility. For example, you can't deliver papers until you have printed them, so if the diagram showed that sequence, it would be wrong. The conclusion is that there is no single right solution, but a diagram can be said to be wrong if it violates logic.

The network for the yard project could get a lot more complicated. You could have *edge front* sidewalk and edge back sidewalk. You could talk about trimming around trees in both front and back, and so on. But there is no need to make it too complicated. We don't usually try to capture *exactly* how we will do the work, just the gist of it.

> **It is hard to tell whether a network is absolutely correct, but it can be said to be wrong if logic is violated.**

The next step is to figure out how long it will take to do the job. Time estimates for each task are made by using history—remembering how long each activity has taken in the past. Remember, though, that the estimate is valid only for the individual who is going to do the task. If my daughter, who is six-

teen, does the lawn mowing using a push mower, it will probably take less time than for my son, who is only twelve. In the following chapter, we see how to find the critical path through the network, so we can know how long it will take.

■■■

Key Points to Remember

Project management is not just scheduling.

Arrow diagrams allow an easier assessment of the impact of a slip on a project than is possible with Gantt charts.

Schedule at a level of detail that can be managed.

No task should be scheduled with a duration much greater than four to six weeks. Subdivide longer tasks to achieve this objective. Software and engineering tasks should be divided even further, to durations not exceeding one to three weeks.

■■■

Exercise . . . .

For the following WBS (Figure 5-5), draw an arrow diagram. One solution is shown in the Answers section.

Figure 5-5. WBS to clean room.

Producing a Workable Schedule

nce a suitable network has been drawn, with durations assigned to all activities, it is necessary to determine where the longest path is in the network and to see whether it will meet the target completion date. Since the longest path through the project determines *minimum* project duration, any activity on that path that takes longer than planned will cause the end date to slip accordingly, so that path is called the *critical path.*

Schedule Computations

Normally, you would let a computer do these computations for you, so you may wonder why it is necessary to know how to do them manually. My belief is that unless you know how the computations are done, you do not fully understand the meanings of float, early and late dates, and so on. Further, you can easily fall prey to the *garbage-in, garbage-out* malady. So here is a brief treatment of how the calculations are done by the computer. (For most schedules, the computer has the added bonus of converting times to calendar dates, which is no easy task to do manually.)

First, consider what we want to know about the project. If it

starts at some *time = zero,* we want to know how soon it can be finished. Naturally, in most actual work projects, we have been told when we must be finished. That is, the end date is dictated. Furthermore, the start date for the job is often constrained for some reason: resources won't be available, specs won't be written, or another project won't be finished until that time. So scheduling usually means trying to fit the work between two fixed points in time. Whatever the case, we still want to know how long the project will take to complete; if it won't fit into the required time frame, then we will have to do something to shorten the critical path.

Failure to consider resource allocation in scheduling almost always leads to a schedule that cannot be achieved.

In the simplest form, network computations are made for the network on the assumption that activity durations are exactly as specified. However, activity durations are a function of the level of resources applied to the work, and if that level is not actually available when it comes time to do the work, then the scheduled dates for the task cannot be met. It is for this reason that network computations must ultimately be made with resource limitations in mind. Another way to say this is that *resource allocation* is necessary to determine what kind of schedule is actually *achievable!* Failure to consider resources almost always leads to a schedule that cannot be met.

Initial schedule computations are made assuming that unlimited reeources are available. This yields the best-case solution.

Still, the first step in network computations is to determine where the critical path is in the schedule and what kind of latitude is available for noncritical work, under *ideal conditions.* Naturally, the ideal situation is one in which unlimited resources are available, so the first computations made for the network are done without consid-

eration of resource requirements. It is this method that is
described in this chapter, and resource allocation methods are
deferred to scheduling software manuals, as I said previously.

Network Rules

In order to compute network start and finish times, only two
rules apply to all networks. These are listed following as rules 1
and 2. Other rules are sometimes applied by the scheduling soft-
ware itself. These are strictly a function of the software and are
not applied to all networks.

Rule 1.   Before a task can begin, all tasks preceding it
            must be completed.

Rule 2.   Arrows denote logical precedence.

Basic Scheduling Computations

Scheduling computations are illustrated using the network in Fig-
ure 6-1. First, let us examine the node boxes in the schedule.
Each has the notation *ES, LS, EF, LF,* and *DU.* These mean:

ES = Early Start

LS = Late Start

EF = Early Finish

LF = Late Finish

DU = Duration (of the task)

Forward-Pass Computations

Consider a single activity in the network, such as picking up
trash from the yard. It has a duration of fifteen minutes. Assum-
ing that it starts at **time** = zero, it can finish as early as fifteen
minutes later. Thus, we can enter 15 in the cell labeled *EF.*

Putting gas in the mower and weed whacker takes only five
minutes. The logic of the diagram says that both of these tasks

Figure 6-1. Network to illustrate computation methods.
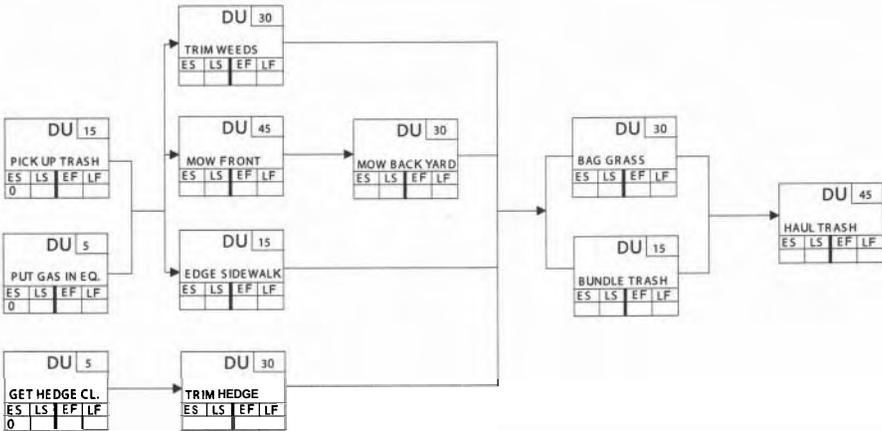
must be completed before we can begin trimming weeds, cutting the front grass, and edging the sidewalk. The cleanup task takes fifteen minutes, whereas the gas activity takes only five minutes. How soon can the following activities start? Not until the cleanup has been finished, since it is the longest of the preceding activities.

In fact, then, the Early Finish for cleanup becomes the Early Start for the next three tasks. It is always true that the *latest Early Finish* for preceding tasks becomes the *Early Start* for subsequent tasks. That is, the longest path determines how early subsequent tasks can start.

Following this rule, we can fill in Earliest Start times for each task, as shown in Figure 6-2. This shows that the project will take a total of 165 minutes-to complete, if all work is conducted exactly as shown. We have just performed what are called *forward-pass computations* to determine Earliest Finish times for all activities. Computer pro-

> The Earliest Start for a task is the latest Late Finish of preceding tasks. That is, the longest path determines the earliest that a following task can be started.

Figure 6-2. Diagram with EF times filled in.
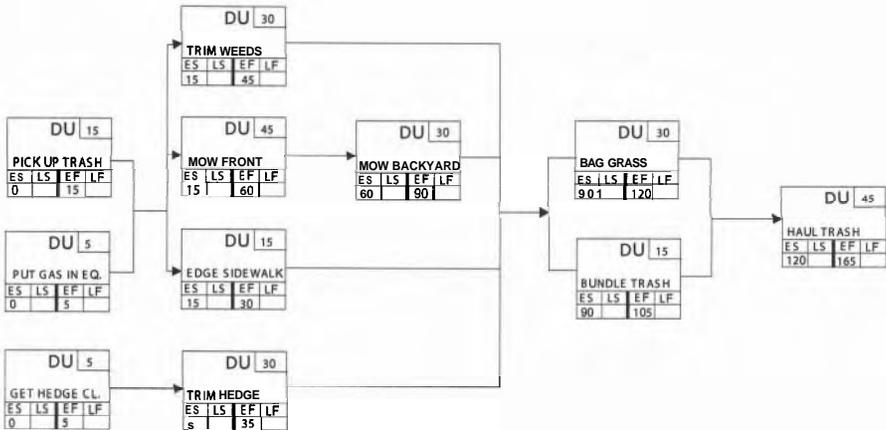
grams do exactly the same thing and additionally convert the times to calendar dates, making quick work of the computations.

> RULE: When two or more activities precede another activity, the earliest time when that activity can be started is the larger of the durations of the activities preceding it.

> NOTE: The time determined for the end or final event is the earliest finish for the project in working time. Once weekends, holidays, and other breaks in the schedule are accounted for, the end date may be considerably later than the earliest finish in working time.

Backward-Pass Computations

A backward pass is made through the network to compute the *latest start and latest finish times* for each activity in the network. To do that, we must decide how late the project can finish. By convention, we generally don't want a project to end any later than its earliest possible completion. To stretch it out longer would be inefficient.

We also won't insist (for now) that the project end earlier than the earliest possible finish calculated in the previous steps. If we want to finish earlier, we will have to redraw the network or

shorten some activities (by applying more resources or working more efficiently, for example). For now, we will accept the 165-minute working time and let it be the *Latest Finish* for the project.

If Hauling Away Trash has a Late Finish of 165 minutes and has a duration of forty-five minutes, what is the latest that it could start? Clearly, if we subtract 45 from 165, we have 120 minutes, which is the Latest Start for the task. Proceeding in this manner, we get LS times for Bagging Grass and Bundling Clippings of 90 and 105 minutes, respectively. One of these two numbers must be the LF time for each of the preceding activities. Which one?

When doing backward-pass calculations, always use the smallest number for the LF of previous activities.

Well, assume we try 105 minutes. If we do that, it would say that Bagging Grass could start as late as 105 minutes, since as soon as preceding tasks are finished subsequent tasks can begin.

But if we add 30 minutes for Bagging to the 105-minute ES time, we will finish at 135 minutes, which is later than the 120 minutes previously determined, and we will miss the 165-minute end time for the project.

When an activity has no float, it is called *critical*, since failure to complete work as scheduled will cause the end date to slip.

Therefore, when we are doing *backward-pass calculations,* the *Latest Finish* for a preceding task will always be the *smallest* of the *Late Start* times for the subsequent tasks. (A simpler way to say this is: always use the smallest number!)

> **RULE:** When two or more activities follow another, the latest time when that (preceding) activity can be achieved is the smaller of the times.

Now examine the path in Figure 6-3 that includes activities highlighted by bold lines. Each activity has the same ES/LS and EF/LF times. There is no float on this path. By convention, an

activity with no float is called *critical,* and a total path with no float is called the *critical path,* which means that if any of the work on this path falls behind schedule, then the end date will slip accordingly. All of the activities that have ES/LS or EF/LF times that differ are said to have float. For example, Trim Weeds has an ES time of fifteen minutes and an LS time of sixty minutes, giving it forty-five minutes of float.

The final network is shown in Figure 6-3. Note that some tasks have the same EF and LF times, as well as the same ES and LS times. These tasks are on the *critical path.* In Figure 6-3, they are shown with bold outlines, to indicate exactly where the critical path lies.

The critical path activities have no latitude. They must be completed as scheduled, or the entire project will take longer than 165 minutes. Knowing where the critical path is tells a manager where his attention must be applied. The other tasks have latitude, which is called *float.* This does not mean that they can be ignored, but they have less chance of delaying the project if they encounter problems. The Edge Sidewalk task, for example, has an ES time of fifteen minutes and an LS time of seventy-five. The difference between the two is sixty minutes, which is the float for the task.

What good is the float? Well, we know we can start the task as late as seventy-five minutes into the job and still finish the project on time. If your son is doing this task, he can watch a sixty-minute television program during that time and still get his Edging task done on time.

Remember, too, that the times are all *estimates.* This means that tasks might take more or less than the scheduled time. So long as they do not take longer than the scheduled time plus the available float time, the job can be completed on time. Critical tasks, which have no float, must be managed in such a way that they take the scheduled time. This is usually done by adjusting the resources (effort)applied, either by assigning more resources or working overtime (increasing resources in either case).

This is not always possible. Applying overtime often increases

Figure 6-3. Diagram showing critical path.
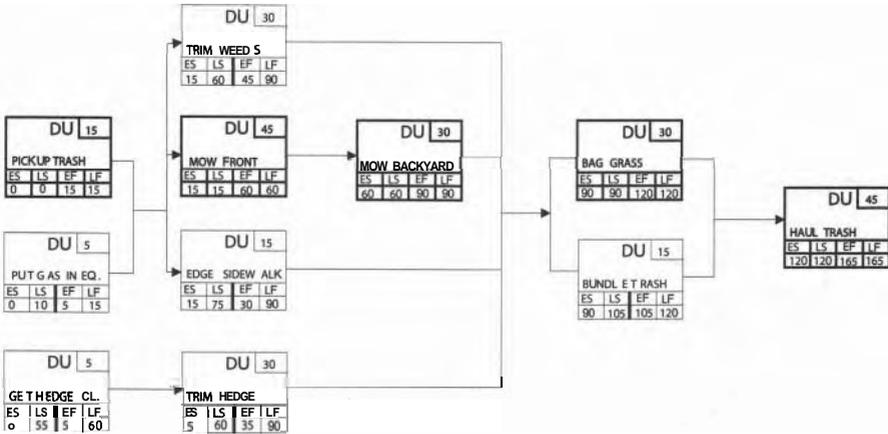
errors, leading to rework, which may mean that you don't get the job done any faster than if you had just worked a normal schedule. Furthermore, there is always a point of diminishing returns when you add bodies to a task. At some point, they just get in each other's way, actually slowing work down rather than speeding it. Note that overtime should be kept in reserve in case of problems, so it is never a good idea to schedule a project so that overtime needs to be worked just to meet the original schedule.

It is bad practice to schedule a project so that overtime is required to meet the schedule, since if problems are encountered, it may not be possible to work more overtime to solve them.

Another point of great importance: all members of the project team should be encouraged to keep float times in reserve as insurance against bad estimates or unforeseen problems. People tend to wait until the latest possible start time to start a task; then, when problems occur, they miss the end date. If there is no float left, when the task takes longer than originally planned, it will impact the end

date for the entire project, since once a task runs out *of float,* it
becomes part of the critical path! In fact,
the true meaning of the word critical is
that there is no float. The task must be
done on time.

Using the Network to
Manage the Project

As I have indicated previously, the point of
developing a **CPM** diagram is to use it to
manage the project. If this is not done, scheduling is simply a
worthless exercise. So here are some pointers that I have found
helpful in managing my own jobs:

Once you have used up the float on a task, i t becomes part of the critical path.

▶ Try to stay on schedule. It is always harder to catch up than
   to stay on target to begin with.

▶ Keep float in reserve in case of unexpected problems or bad
   estimates.

▶ Apply whatever effort is needed to keep critical tasks on
   schedule. If a task on the critical path can be finished ahead
   of schedule, do *it!* Then start the next task.

▶ Avoid the temptation to perfect everything—that's what the
   next-generation product or service is all about. Note: I did
   not say it is okay to do the job sloppily or that you shouldn't
   do your best work. I said don't be tempted to make it per-
   fect. By definition, you will never reach perfection.

▶ Estimates of task durations are made assuming the applica-
   tion of certain people. If someone else is actually used, you
   may have to adjust durations accordingly. This is especially
   true if the person is less skilled than the intended resource.

▶ This was stated in chapter 5 but is repeated because of its
   importance: No task should be scheduled with a duration
   much greater than four to six weeks. If you do, people tend

to have a false sense of security and put off starting, under the assumption "I can always make up one day." By the time they start, they often have slipped several days and find that they cannot finish as scheduled. We say that they back-end *load the* task by pushing all the effort toward the back end. If a task has a duration greater than six weeks, it is a good idea to subdivide it, creating an artificial break if necessary. Then review progress at that point. That **will** help keep it on target.

► If the people doing the work did not develop the network, explain it to them and show them the meaning of float. Don't hide it from them. However, give them a bar chart to work to—it is much easier to read a bar chart than a network diagram. Show them that if they use up float on a given task, then the following tasks may become critical, leaving the people who must do those activities feeling really stressed.

► It is possible to shorten a task by adding resources, reducing its scope, doing sloppy (poor-quality)work, being more efficient, or changing the process by which the work is done. With the exception of doing sloppy work, all of the methods may be acceptable. A reduction in scope must be negotiated with your customer, of course.

■ Scheduling is done initially on the assumption that you **will** have the resources you planned on having. If people are shared with other projects, or if you plan to use the same person on several tasks, you may find that you have her overloaded. Modern software generally warns you that you have overloaded your resources and may be able to help you solve the problem.

Converting Arrow Diagrams to Bar Charts

While an arrow diagram is essential to do a proper analysis of the relationships between the activities in a project, the best working tool is the bar chart. The people doing the work will find it much easier to see when they are supposed to start and finish their jobs if you give them a bar chart. The arrow diagram in Figure 6-3 has

been portrayed as a bar chart in Figure 6-4, making use of what was learned about the schedule from the network analysis.

Figure 6-4. Bar chart schedule for yard project.

PICKUP TRASH
PUT GAS IN EQUIPMENT
GET OUT HEDGE CLIPPER
TRIM WEEDS
MOW FRONT LAWN
EDGE SIDEWALK
TRIM HEDGE
MOW BACKYARD
BAG GRASS & TRASH
BUNDLE HEDGE CLIPPINGS
HAUL AWAY TRASH

TIME, MINUTES

0   25   50   75   100   125   150   175

TASK **WITH** FLOAT   CRITICAL TASK
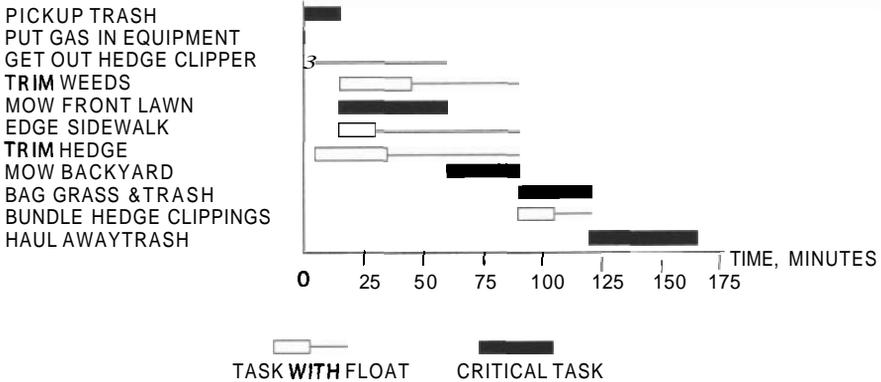
Note that the critical path in the bar chart is shown as solid black bars. Bars with float are drawn hollow with a line trailing to indicate how much float is available. The task can end as late as the point at which the trailing line ends.

This is fairly conventional notation. Scheduling software always allows you to print a bar chart, even though a CPM network is used to find the critical path and to calculate floats. One caution: many programs display the critical path in red on a color monitor and often color started tasks with green or blue. When these bars are printed on a black-and-white printer, all of them may look black, implying that they are all critical, confusing the people trying to read them. It is usually possible to have the computer display shading or cross-hatching instead of color so that when they are printed in black-and-white, there **will** be no ambiguity.

Assigning Resources to Tasks

I have already said that the first step in developing a schedule is to assume that you have unlimited resources, because this is the

best situation you can ever assume, and if you can't meet your project completion date with an unlimited resource schedule, you may as well know it early. However, once you have determined that the end date can somehow be met, you now must see whether your unlimited resource assumption has overloaded your available resources.

Normally you will find that you have people double- and triple-scheduled, which clearly won't work. These kinds of resource overloads can be resolved only by using computer software, except for very simple schedules. This is where the software really excels, and yet estimates are that only a few percent of all the people who purchase software actually use it to level resources.

Consider the small schedule in Figure 6-5. It contains only four tasks. Two are critical, and two have float. Task **A** requires two workers if it is to be completed in three weeks, and tasks B

Figure 6-5. Schedule with resources overloaded.
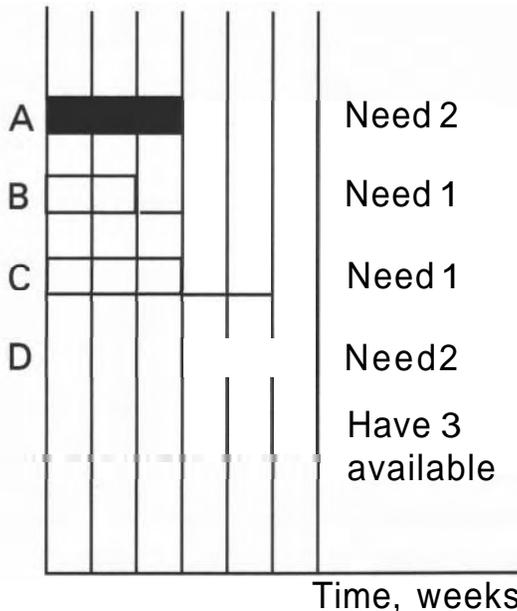
and C need one person each. When it comes time to do the project, however, you find that there are only three workers available. How did this happen?

It is possible that no more than three people were ever available, but because you followed the rule to schedule in parallel tasks that could logically be done in parallel, you inevitably overloaded your people. It is also possible that, when the plan was constructed, four workers were available but that one has since been assigned to another job that has priority over yours.

Whatever the reason, this schedule won't work unless something is changed. There are a number of possibilities. There are three areas to examine. You should first see whether any task has enough float to allow it to be delayed until resources become available. In this particular example, it turns out that this is possible. The solution is shown in Figure 6-6.

Of course, this solution is a nice textbook example that just happens to work out. It is never so easy in a real project. Notice that task C has enough float that it can slide over and wait until activity B is finished. But what usually happens is that task C runs out of float before B is completed. Also, assume that task D needs three people, rather than two. **As** you can see, this complicates the situation considerably. This is shown in Figure 6-7.

Since this is the typical situation, we must be prepared to handle it. There are two more places to look for help. The first is the functional relationship among the variables:

$$C = f(P, T, S)$$

You should ask whether you can reduce scope, change the time limit, or reduce performance. Usually performance is not negotiable, but the others may be. For example, sometimes you can reduce scope, and the project deliverable will still be acceptable to the client. Of course, if you can get another person for a short time, you won't have to consider reducing scope or performance. So you go shopping.

You ask the manager who "owns" the resources whether she can provide another person. She says sadly that she cannot and

Figure 6-6. Schedule using float to level resources.

Time, weeks
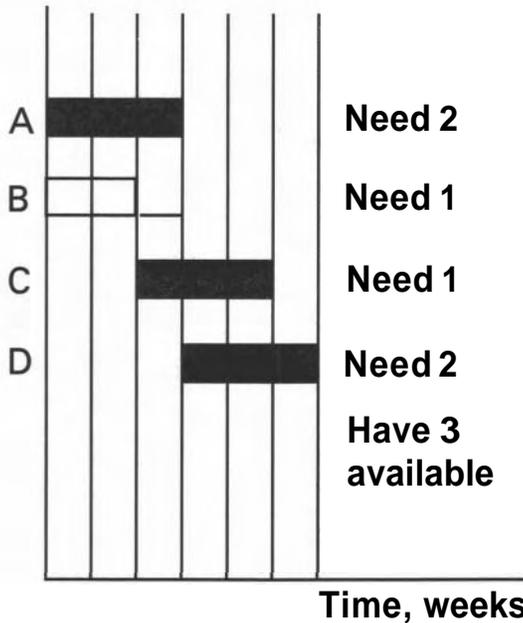
that she was even considering trying to take back another of the three she has already given you. Somehow you convince her not to do this. You then ask the project sponsor if it is okay to reduce scope. It is not.

It is also not okay to reduce performance. Nor can you find a contract employee in time to do the job. You are between a rock and a hard place. So you now ask whether there is another process that could be used to do the work. For example, if you can spray paint a wall instead of using a roller, it may go much faster.

Suppose you try this and again you come up empty-handed. You decide the only thing left to do is resign your job. You never really wanted to be a project manager, anyway. But wait. Perhaps there is something else you can do.

Think back to what I said earlier. You use up all the float on C, and it is now a critical-path task. When you tell your software to level resources, it wants to know whether you want to schedule

**Figure 6-7. Schedule with inadequate float on C
to permit leveling.**
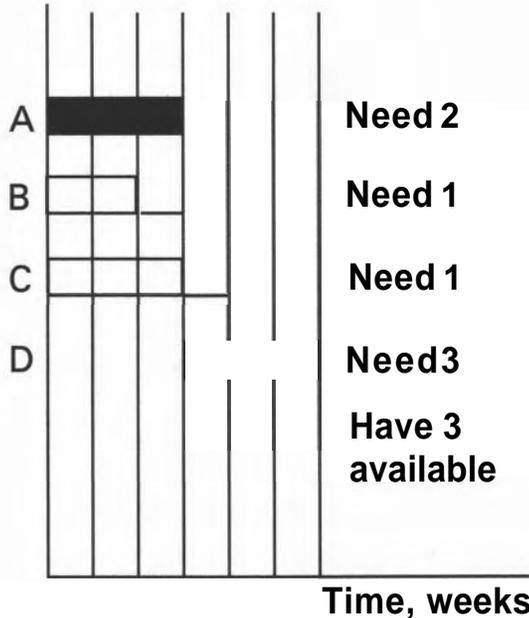
within the available float (or slack, as it is also called). If you say yes, as soon as a task runs out of float, it won't move over any further. This is also called *time-critical* resource leveling, because time is of the essence for your project. (It always is!)

However, suppose you answer "no" to the question, "Do you want to level within the available slack?" In this case, you are telling the software to continue sliding tasks over until resources become available, even if it means slipping the end date. (This is called *resource-critical* leveling.) When you try this with our example schedule, you arrive at the solution shown in Figure 6-8. Not bad, unless you can't live with the slip.

In fact, sometimes the slip is so bad that it seems almost ridiculous. Your project was originally going to end in December of the current year. Now the software says it is so starved for resources that it will end in the year 2013! Ridiculous! What good is a schedule that goes out that far?

Figure 6-8. Schedule under resource-critical conditions.
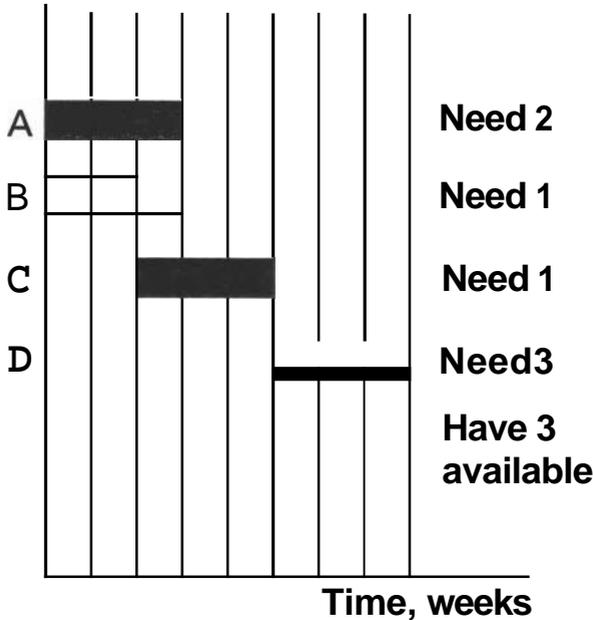
It can be used to bring the issue to everyone's attention. It shows the impact of inadequate resources and forces a trade-off as described earlier—that is, if everyone believes your schedule in the first place. I have just had an experience with a fellow who said that he didn't believe the schedules in the first place, because he thought they were always unrealistic, so an unrealistic schedule subjected to fancy calculations didn't prove anything to him.

I'm sure that's true. However, if people are willing to accept the limitations of what we are doing when we plan a project, this is at least a way of showing the limitations you face. Everyone must understand that estimating is *guessing* as is true of market and weather forecasting, neither of which has a stellar record. Moreover, all activities are subject to variation, as I have pointed out. If people don't understand this, then I suggest you turn in your project manager's hat for a better job.

Resource Availability

A major factor in dealing with resource allocation is the availability of each person to do project work. One guideline that industrial engineers follow is that no person is available to work more than 80 percent of the time. Considering an eight-hour day, that means 6.4 hours a day available for work, and prudence says to just make it six hours. The 20 percent lost availability goes to three factors called PFD. P means personal—every individual must take breaks. F is for fatigue—you lose productive time as people get tired. And D means delays—people lose time waiting for inputs from others, supplies, or instructions on what to do.

Experience shows, however, that the only people who are available to work even 80 percent of the time are those whose jobs tie them to their workstations. This is true for factory workers and others who do routine jobs like processing insurance claims (and even these people move around). With knowledge workers, you never get 80 percent of a day in productive work. The figure is usually closer to 50 percent, and it may be lower! One company that I know of did a time study in which people logged their time every hour for two weeks, and they found that project work accounted for only 25 percent of their time. The rest went to meetings, nonproject work that had to be done, old jobs that were finished long ago but came back to the person who originally worked on them, work on budgets for the next year, customer support, and on and on.

Most software programs allow you to specify the number of *working hours* needed for a task and the percentage of a day that a person **will** work on the task; the software then translates that into calendar time. So, as an example, if a person is working on your project only half time and the task she is doing is supposed to take twenty hours of actual working time, then it will be a week (or more) before she finishes it.

It is especially important that you know the availability of people to do project work, or you **will** produce schedules that are worse than useless. I say worse, because they will be mislead-

ingly short, and they will wreak havoc with your organization. Do a time study to determine the number, then use it. And if people don't like the fact that a lot of time is being lost to nonproject activities, then correct the problem by removing those disruptive activities.

The usual solution is that people must work overtime to get their project work done, because of all the disruptions that occur during the day. The problem is that studies have found that overtime has a very negative impact on productivity. So it is a losing battle. Short-term overtime is fine, but long spans just get organizations into trouble.

■■■
Key Points to Remember

You should ignore resource limitations when you begin developing a schedule. If two tasks can logically be done in parallel, draw them that way.

The critical path is the one that is longest and has no float. Note that you can have a project with a longest path that is not critical because it has float.

Nobody is available to do productive work more than 80 percent of a workday. You lose 20 percent to personal time, fatigue, and delays.
■■■

Exercise . . . .

For the network in Figure 6-9, calculate the early and late times and the float available on noncritical activities. Which activities form the critical path? Answers are in the Answers section at the back of the book.

Figure 6-9. Network for exercise.
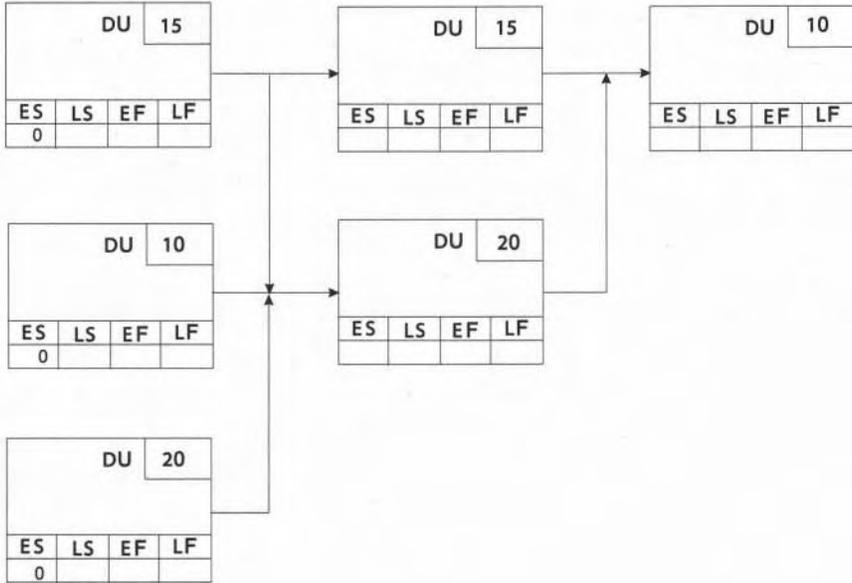
Project Control
and Evaluation

E very step taken up to now has been for one purpose—to achieve control of the project. This is what is expected of a project manager—that organization resources be managed in such a way that critical results are achieved.

However, there are two connotations to the word *con-trol,* and it is important that we use the one that is appropriate in today's world. One meaning of *control* is that of domination, power, command. We control people and things through the use of that power. When we say "Jump," people ask, "How high?" At least they used to. It doesn't work that well today.

I have previously discussed the fact that project managers often have a lot of responsibility but little authority. Let's examine that and see whether it is really a problem.

I have asked several corporate officers (presidents and vice presidents),"Since you have a lot of authority, does that authority guarantee that people will do what you want done?"

Uniformly, they answer, "No."

"What does get them to do what you want done?"

"Well, in the end analysis, they have to want to do it," they say.

"Then what does your authority do for you?" I ask.

"Well, it gives me the right to exercise sanctions over them, but that's all."

So we find that having authority is no guarantee that you will be able to get people to do your bidding. In the end, you have to get them to do it willingly, and that says you have to understand the motivations of people so that you can *influence* them to do what needs to be done.

There are two kinds of authority: One is power over people, and the other is the ability to make decisions and to act unilaterally.

A second kind of authority has to do with taking actions unilaterally — that is, without having to get permission first. In this sense of the word, we do have a lot of organizational problems. I meet project managers who have project budgets in the millions of dollars (as much as $35 million in one case), yet who must have all expenditures approved. If a project plan and budget have been approved before the work was started, and if the project manager is spending within the approved limits of the plan, why should she have to get more signatures for approved expenditures? Only if a deviation from the plan is going to result should more signatures be needed, and then the plan should be revised to reflect those changes.

Consider the messages being sent to these managers. On the one hand, they are being told, "We trust you to administer $35 million of our money." On the other hand, they are told, "But when you spend it, you must have every expenditure approved by someone of higher authority." One is a positive message: We trust you. The other is negative. Which do you think comes through loud and clear? You bet! The negative.

A negative message always takes priority over a positive one.

Interestingly, we complain that people in organizations won't take more responsibility for themselves; then we treat them as though they are irresponsible and wonder why they don't behave responsibly!

The first meaning of control has a power connotation. Another meaning is summed up by the highlighted definition. This definition was introduced in an earlier chapter. Control is the act of comparing progress to plan so that corrective action can be taken when a deviation from planned performance occurs. This definition implies the use of *infonnation* as the primary ingredient of control, rather than power. Thus, we talk about management information systems, and, indeed, these are the essence of what is needed to achieve control in projects.

con•trol: to com-pare progress against plan so that corrective action can be taken when a deviation occurs

Unfortunately, many organizations have management information systems that are good for tracking inventory, sales, and manufacturing labor but not for tracking projects. Where such systems are not in place, you will have to track progress manually.

Achieving Team Member Self-Control

Ultimately, the only way to control a project is for every member of the project team to be in control of his or her own work. A project manager can achieve control at the *macro* level only if it is achieved at the *micro* level. However, this does not mean that you should practice micro-managing! It actually means that you should set up conditions under which every team member can achieve control of his or her own efforts.

To do this requires five basic conditions. These are shown separately. To achieve self-control, team members need:

▶ A clear definition of what they are supposed to be doing, with the purpose stated.

▶ A personal plan for how to do the required work.

▶ Skills and resources adequate to the task.

▶ Feedback on progress that comes directly from the work itself.

▶ A clear definition of their authority to take corrective action when there is a deviation from plan (and it cannot be zero!).

The first requirement is that every team member be clear about what her objective is. Note the difference between tasks and objectives, which was discussed in chapter **3.** State the objective and explain to the person (if necessary) what the purpose of the objective is. This allows the individual to pursue the objective in her own way.

The second requirement is for every team member to have a personal plan on how to do the required work. Remember, if you have no plan, you have no control. This must apply at the individual as well as at the overall project level.

The third requirement is that the person have the skills and resources needed for the job. The need for resources is obvious, but this condition suggests that the person may have to be given training if she is lacking necessary skills. Certainly, when no employee is available with the required skills, it may be necessary to have team members trained.

The fourth requirement is that the person receive feedback on performance that goes directly to her. If such feedback goes through some roundabout way, she cannot exercise self-control. To make this clear, if a team member is building a wall, she can measure the height of the wall, compare it to the planned performance, and know whether she is on track.

The fifth condition is that the individual must have a clear definition of her authority to take corrective action when there is a deviation from plan, and it must be greater than zero authority! If she has to ask the project manager what to do every time a deviation occurs, the project manager is still controlling. Furthermore, if many people have to seek approval for every minor action, this puts a real burden on the project manager.

Characteristics of a Project Control System

The control system must focus on project objectives, with the aim of ensuring that the project mission is achieved. To do that, the control system should be designed with these questions in mind:

▶ What is important to the organization?

▶ What are we attempting to do?

▶ Which aspects of the work are most important to track and control?

▶ What are the critical points in the process at which controls should be placed?

Control should be exercised over what is important. On the other hand, what is controlled tends to become important. Thus, if budgets and schedules are emphasized to the exclusion of quality, only those **will** be controlled. The project may well come in on time and within budget, but at the expense of quality. Project managers must monitor performance carefully to ensure that quality does not suffer.

Taking Corrective Action

A control system should focus on response—if control data do not result in action, then the system is ineffective. That is, if a control system does not use deviation data to *initiate corrective action,* it is not really a control system but simply a monitoring system. If you are driving and realize that you have somehow gotten on the wrong road but do nothing to get back on the right road, you are not exercising control.

One caution here, though. I once knew a manager whose response to a deviation was to go into the panic mode and begin micro-managing. He then got in the way of people trying to solve the problem and actually slowed them down. Had he left them alone, they would have solved their problem much faster.

Timeliness of Response

The response to control data must be timely. If action occurs too late, it will be ineffective. This is frequently a serious problem. Data on project status are sometimes delayed by four to six weeks, making them useless as a basis for taking corrective action. Ideally, information on project status should be available on a *real-time* basis. In most cases, that is not possible. For many projects, status reports that are prepared weekly are adequate.

Ultimately, you want to find out how many hours people *actually* work on your project and compare that figure to what was *planned* for them. This means that you want accurate data. In some cases, people fill out weekly time reports without having written down their working times daily. That results in a bunch of fiction, since most of us cannot remember with any accuracy what we did a week ago.

As difficult as it may be to do, you need to get people to record their working times daily so that the data will mean something when you collect them. What's in it for them? Perhaps nothing. Perhaps future estimates will be better as a result of col-

> **When people fill out time reports weekly, without writing down what they did *daily,* they are making up fiction. Such made-up data are almost worse than no data at all.**

lecting accurate information on this project. In any case, you need accurate data, or you may as well not waste your time collecting them.

When information collection is delayed for too long, the manager may end up making things worse, instead of better. Lags in feedback systems are a favorite topic for systems theorists. The government's attempts to control recessions and inflation sometimes involve long delays, as a result of which the government winds up doing the exact opposite of what should have been done, thereby making the economic situation worse.

There is one point about control that is important to note. If every member of the project team is practicing proper control

methods, then reports that are prepared weekly are just checks and balances. This is the desired condition.

Designing the Right System

One system is not likely to be correct for all projects. It may need to be scaled down for small projects and beefed up for large ones. Generally, a control system adequate for a large project will overwhelm a small one with paperwork, while one that is good for small projects won't have enough "clout" for a big project.

Practicing the KISS *Principle*

KISS stands for "Keep it simple, stupid!" The smallest control effort that achieves the desired result should be used. Any control data that are not essential should be eliminated. However, as was just mentioned, one common mistake is to try to control complex projects with systems that are *too simple!*

To keep control simple, it is a good idea to check periodically that reports that are generated are actually being used for something by the people who receive them. We sometimes create reports because we believe the information in them should be useful to others, but if the recipients don't actually use

> No problem *is* so big or *so* complicated that it can't be run away from.
>
> —Charlie Brown (Charles Schultz, *Peanuts*)

it, we kid ourselves. To test this point, send a memo with each report telling people to let you know whether they want to receive future reports; if you do not hear from them, their names will be removed from the distribution. You may be surprised to find that *no one* uses some of your reports. Those reports should be dropped completely.

Project Review Meetings

There are two aspects to project control. One can be called maintenance, and the other aims at improvement of performance. The maintenance review just tries to keep the project on track. The

improvement review tries to help project teams improve performance. Three kinds of reviews are routinely conducted to achieve these purposes. They are:

■ Status reviews

▶ Process or lessons-learned reviews

■ Design reviews

Everyone should do status and process reviews. Design reviews, of course, are appropriate only if you are designing hardware, software, or some sort of campaign, such as a marketing campaign.

A status review is aimed at maintenance. It asks where the project stands on the PCTS measures that we have used throughout this book. Only if you know the value of all four of these can you be sure where you are. This is the subject of the next chapter.

Process means the way something is done, and you can be sure that process always affects task performance. That is, *how* something is done affects the outcome. For that reason, process improvement is the work of every manager. How this is done is covered in the next section.

Project Evaluation

As the dictionary definition says, to evaluate a project is to attempt to determine whether the overall status of the work is acceptable, in terms of intended value to the client once the job is finished. Project evaluation appraises the progress and performance of a job compared to what was originally planned. That evaluation provides the basis for management decisions on how to proceed with the project. The evaluation must be credible in the eyes of everyone affected, or decisions based on it

e•val•u•ate: to determine or judge the value or worth of

—*The Random* House *Dictionary*

will not be considered valid. The primary tool for project evaluation is the *project process review,* which is usually conducted at major milestones throughout the life of the project.

Purposes of Project Evaluation

Sports teams that practice without reviewing performance may get really *good* at playing very *badly.* That is why they review game films—to see where they need to improve. In other words, the purpose of a review is to learn lessons that can help the team to avoid doing things that cause undesired outcomes and to continue those that help. The review should be called a *lessons-learned* or *process* review.

I have deliberately avoided the word *audit,* because nobody likes to be audited. Historically, an audit has been designed to catch people doing things they shouldn't have done so that they can be penalized in some way. If you go around auditing people, you can be sure they **will** hide from you anything they don't want you to know, and it is those very things that could help the company learn and grow.

As Dr. W. Edwards Deming has pointed out in his book, *Out of the Crisis,* there are two kinds of organizations in this world today—those that are getting better and those that are dying. An organization that stands still is dying. It just doesn't know it yet.

The reason? The competition is not sitting by idly. It is doing new things, some of which may be better than yours. If you aren't improving, you will be passed by, and soon you won't have a market.

The same is true of every part of an organization. You can't suboptimize, improving just manufacturing. You have to improve every department, and that includes how you run projects.

Good management of projects can give you a competitive advantage.

In fact, good project management can give you a real competitive advantage, especially in product development. If you are sloppy in managing your projects, you don't have good control of development

costs. That means that you have to either sell a lot of product or charge large margins to cover your development costs so that the project is worth doing in the first place. If you can't sell a lot of widgets, then you have to charge the large margin.

If your competitor, on the other hand, has good cost control, it can charge smaller margins and still be sure that it recovers its investment and makes money. Thus, it has a competitive advantage over you because of its better control of project work.

Additionally, in order to learn, people require feedback, like that gained by a team from reviewing game films. The last phase of a project should be a final process review, conducted so that the management of projects can be improved. However, such a process review should not be conducted only at the end of the project. Rather, process reviews should be done at major milestones in the project or every three months, whichever comes first, so that learning can take place as the job progresses. Furthermore, if a project is getting into serious trouble, the process review should reveal the difficulty so that a decision can be made to continue or terminate the work.

In order to learn, we must have *feedback*. Furthermore, we tend to learn more from mistakes than from successes, painful though that may be to admit.

Following are some of the general reasons for conducting periodic project process reviews. You should be able to:

.Improve   project performance together with the management of the project.

■ Ensure that quality of project work does not take a back seat to schedule and cost concerns.

■ Reveal developing problems early so that action can be taken to deal with them.

■ Identify areas where other projects (current or future) should be managed differently.

> ► Keep client(s) informed of project status. This can also help ensure that the completed project **will** meet the needs of the client.

> ► Reaffirm the organization's commitment to the project for the benefit of project team members.

Conducting the Project Process Review

Ideally, a project process review should be conducted by an independent examiner, who can remain objective in the assessment of information. However, the process review must be conducted in a spirit of learning, rather than in a climate of blame and punishment. If people are afraid that they will be "strung up" for problems, then they **will** hide those problems if at all possible.

Even so, openness is hard to achieve. In many organizations, the climate has been punitive for so long that people are reluctant to reveal any less-than-perfect aspects of project performance. Dr. Chris Argyris, in his book, *Overcoming Organizational Defenses,* has described the processes by which organizations continue ineffective practices. All of them are intended to help individuals "save face" or avoid embarrassment. In the end, they also prevent organizational learning.

Process reviews conducted as witch-hunts will produce witches.

Two questions should be asked in the review. The first is, "What have we done well so far?," and the second is, "What do we want to improve (or do better) in the future?" Notice that I am not asking, "What have we done badly?" That question serves only to make everyone defensive, because they assume that you will punish them for things done wrong. Furthermore, there is always the possibility that nothing has been done wrong, but there is always room to improve.

Finally, the results of the review should be published. Otherwise, the only people in the organization who can take advantage of it are the members of the team just reviewed. If other teams

know what was learned, then they can benefit from that information. In the next section, we look at what the report should contain.

The Process Review Report

A company may decide to conduct process reviews in varying degrees of thoroughness, from totally comprehensive, to partial, to less formal and cursory. A formal, comprehensive process review should be followed by a report. The report should contain as a minimum the following:

> *Current project status.* The best way to do this is to use earned value analysis, as presented in the following chapter. However, when earned value analysis is not used, status should still be reported with as great accuracy as possible.

> ■ *Future status.* This is a forecast of what is expected to happen in the project. Are significant deviations expected in schedule, cost, performance, or scope? If so, the report should specify the nature of the changes.

> ■ *Status of critical tasks.* The report should describe the status of critical tasks, particularly those on the critical path. Tasks that have high levels of technical risk should be given special attention, as should those being performed by outside vendors or subcontractors, over which the project manager may have limited control.

> *Risk assessment.* The report should mention any identified risks that could lead to monetary loss, project failure, or other liabilities.

> *Information relevant to other projects.* The report should describe what has been learned from this process review that can/should be applied to other projects, whether in progress or about to start.

> *Limitations of the process review* The report should mention any factors that may limit the validity of the process

review. Are any assumptions suspect? Are any data missing or perhaps contaminated? Was anyone uncooperative in providing information for the process review?

As a general comment, the simpler and more straightforward a project process review report, the better. The information should be organized so that planned versus actual results can be easily compared. Significant deviations should be highlighted and explained.

● ●
Key Points to Remember

The meaning of control that is important to project managers is the one implying the use of information; comparing progress to plan so that action can be taken to correct for deviations from plan.

The only way a project is really in control is if all team members are in control of their own work.

The effort used to control a project should be worthwhile. You don't want to spend $100 to purchase a $3 battery, for example.

If you take no action in response to a deviation, you have a monitoring system, not a control system.

Project working times must be recorded daily. If people wait a week to capture what they have done, they rely on memory and end up writing down estimates of what they did. Such data are no good for future estimating.

Project evaluation is done to determine whether a project should continue or be canceled. Process reviews also should help the team learn in order to improve performance.
● ●

Project Control Using Earned Value Analysis

ontrol is exercised to achieve project objectives, and we know that there are performance, cost, time, and scope targets that are always important. Furthermore, we have seen that control is exercised by comparing performance to plan and, when deviations or *variances* occur, taking corrective action to bring performance back on target.

As I said in chapter 7, the review that is concerned with maintenance or straightforward project control is the *status* review. This review asks where the project is in terms of all four PCTS variables. Each time progress is reviewed, you must ask these three questions:

1. Where are we (in terms of PCTS)?

2. When there is a deviation, what caused it?

3. What should be done about the deviation?

Note that there are only four actions that can be taken (in response to question 3). These are:

1. Cancel the project.

2. Ignore the deviation.

3. Take corrective action to get back onto the planned progress.

4. Revise the plan to reflect a change in status that can't be corrected.

Sometimes a project gets so far off track that it is no longer viable, and the best thing to do is to cancel it. Of course, this step is not taken lightly, but it should be taken in cases where you are just going to throw good money after bad. Cut your losses and get on with something better.

As for ignoring a deviation, if you can control to within a certain percentage tolerance and you are within those limits, you should usually ignore a deviation unless it shows a trend that **will** definitely eventually take it outside the limits. Otherwise, tweaking may just make the situation worse.

As for taking corrective action, there is no way to tell what this means, as it is specific to each project. Sometimes working people overtime gets a project back on track. Or perhaps you need to add people, or cut scope, or change the process. You must determine what must be done for your project.

In the event that the project is still viable but nothing can be done to get it back on track, you may have to revise the plan. Of course, you can also consider working overtime or reducing scope, since these were not originally called for. What I am really referring to here, however, is a situation in which you cannot recover, and you are revising the plan to show that the costs will increase, the deadline will slip, or some other change to the plan will occur.

> Another day, another zero.
>
> —Alfalfa (Carl.Switzer)
> *Our Gang comedy* series

Measuring Progress

One of the hardest things to do in managing projects is to actually measure progress. When you are following a road map, you monitor the road signs and see whether they agree with your planned route. In well-defined jobs, such as construction projects, it is generally fairly easy to tell where you are. You can measure the height of a brick wall or see whether all the conduit is installed, and so on. That is, you can tell where you are when a part of the work is actually *finished*. When work is poorly defined, and it is only partially complete, however, you have to estimate where you are.

This is especially true of knowledge work — work done with one's head, rather than one's hands. If you are writing software code, designing something, or writing a book, it can be very hard to judge how far along you are and how much you have left to do.

Naturally, if you can't tell where you are, you can't exercise control. And note that use of the word estimate in measuring progress. What exactly is an estimate?

It's a guess.

And so we are guessing about where we are.

Yes. We'll know where we are when we get there. Until we actually arrive, we're guessing.

Does this not sound like something from Alice in Wonderland?

Heavens.

What was that definition of control again? Let's see—compare where you are . . .

How do you know where you are?

We're guessing.

. . . against where you are supposed to be. . . .

How do you know where you're supposed to be?

Oh, that's much easier. The plan tells us.

But where did the plan come from?

It was an estimate, too.

Oh. So if one guess doesn't agree with the other guess, we're

supposed to take corrective action to make the two of them agree, is that it?

That's what this guy Jim Lewis says in his book.

Must be a book on witchcraft and magic.

Well, since it is impossible to know for sure where we are, then perhaps we should just give up on the whole thing and keep running projects by the seat of the pants. Right?

Wrong.

The fact that measures of progress are not very accurate does not justify the conclusion that they shouldn't be used. Remember, if you have no plan, you have no control, and if you don't try to monitor and follow the plan, you definitely don't have control. And if you have no control, there is no semblance of managing. You're just flailing around.

> **The difficulty of measuring progress does not justify the conclusion that it shouldn't be done. You cannot have control unless you measure progress.**

What is important to note, however, is that some projects are capable of tighter control than others. Well-defined work, which can be accurately measured, can be controlled to tight tolerances. Work that is more nebulous (e.g., knowledge work) has to allow larger tolerances. Management must recognize this and accept it. Otherwise, you go crazy trying to achieve **3** percent tolerances. It's like trying to push a noodle into a straight line or nail jelly to a wall.

Measuring Project Performance/Quality

If you think measuring progress is hard, try measuring quality. Were the bolts holding the steel beams together put in properly? Are all the welds sound? How do you tell?

This is the hardest variable to track, and one that often suffers as a consequence. Also, so much attention tends to be focused on cost and schedule performance that quality of the work is often sacrificed. This can be a disaster, in some cases resulting in the fil-

ing of lawsuits against a company for damages that result from poor quality.

Project managers must pay special attention to the quality variable, in spite of the difficulty of tracking it.

Earned Value Analysis

Work quality is most likely to be sacrificed when deadlines are tight. Constant attention is required to avoid this tendency.

It is one thing to meet a project deadline at any cost. It is another to do it for a *reasonable* cost. Project cost control is concerned with ensuring that projects stay within their budgets, while getting the work done on time and at the correct quality.

One system for doing this, called *earned value analysis,* was developed in the 1960s to allow the government to decide whether a contractor should receive a progress payment for work done. The method is finally coming into its own outside government projects, and it is considered the correct way to monitor and control almost any project. The method is also called simply *variance analysis.*

Variance analysis allows the project manager to determine trouble spots in the project and to take corrective action. The following definitions are useful in understanding the analysis:

Cost variance: Compares deviations and performed work.

▶ *Schedule variance:* Compares planned and actual work completed.

BCWS: (Budgeted cost of work scheduled): The budgeted cost of work scheduled to be done in a given time period, or the level of effort that is supposed to be performed in that period.

BCWP: (Budgeted cost of work performed): The budgeted cost of work actually performed in a given period, or the bud-

geted level of effort actually expended. BCWP is also called earned value and is a measure of the dollar value of the work actually accomplished in the period being monitored.

- ACWP: (Actual cost of work performed): The amount of money (or effort) actually spent in completing work in a given period.

Variance thresholds can be established that define the level at which reports must be sent to various levels of management within an organization.

$$\text{Cost Variance} = \text{BCWP} - \text{ACWP}$$
$$\text{Schedule Variance} = \text{BCWP} - \text{BCWS}$$
$$\text{Variance: Any deviation from plan}$$

By combining cost and schedule variances, an integrated cost/schedule reporting system can be developed.

Variance Analysis Using Spending Curves

Variances are often plotted using spending curves. A BCWS curve for a project is presented in Figure 8-1. It shows the cumulative spending planned for a project and is sometimes called a baseline plan.

In the event that software is not available to provide the necessary data, Figure 8-2 shows how data for the curve are generated. Consider a simple bar chart schedule. Only three tasks are involved. Task A involves forty labor-hours per week at an average loaded labor rate of $20 per hour, so that task costs $800 per week. Task B involves 100 hours per week of labor at $30 per how, so it costs $3,000 per week. Finally, task C spends $2,400 per week, assuming sixty hours per week of labor at $40 per hour.

At the bottom of the chart we see that during the first week $800 is spent for project labor; in the second week both tasks A and B are running, so the labor expenditure is $3,800. In the third week, all three tasks are running, so labor expenditure is the sum of the three, or $6,200. These are the weekly expenditures.

Figure 8-1. BCWS curve.
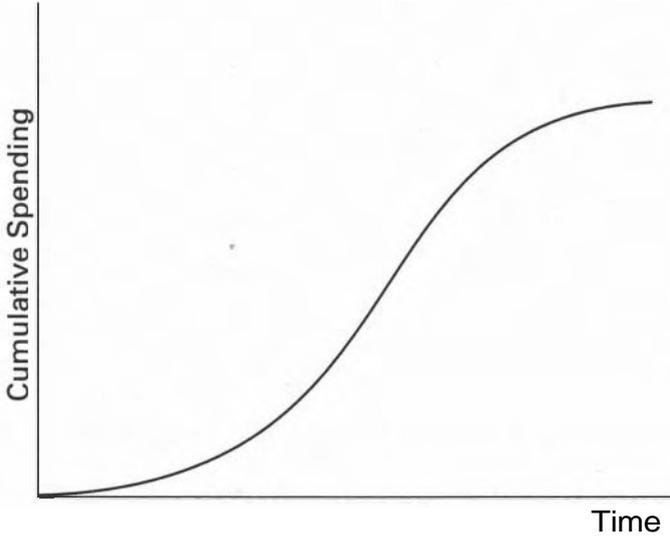
**Figure 8-2. Bar chart schedule illustrating
cumulative spending.**

The *cumulative* expenditures are calculated by adding the cost for each subsequent week to the previous cumulative total. These cumulative amounts are plotted in Figure 8-3. This is the spending curve for the project and is called a BCWS curve. Since it is derived directly from the schedule, it represents *planned performance* and therefore is called a *baseline plan.* Furthermore, since control is exercised by comparing progress to plan, this curve can be used as the basis for such comparisons so that the project manager can tell the status of the program. The next section presents examples of how such assessments are made.

Examples of Progress Tracking Using Spending Curves

Consider the curves shown in Figure 8-4. On a given date, the project is supposed to have involved $40,000 (40K) in labor (BCWS). The actual cost of the work performed (ACWP) is 60K. These figures are usually obtained from Accounting and are derived from all the time cards that have reported labor applied to the project. Finally, the budgeted cost of work performed (BCWP) is 40K. Under these conditions, the project would be behind schedule and overspent.

Figure 8-5 illustrates another scenario. The BCWP and the ACWP curves both fall at the same point, 60K. This means that the project is ahead of schedule but spending correctly for the amount of work done.

Figure 8-3. Cumulative spending for the sample bar chart.
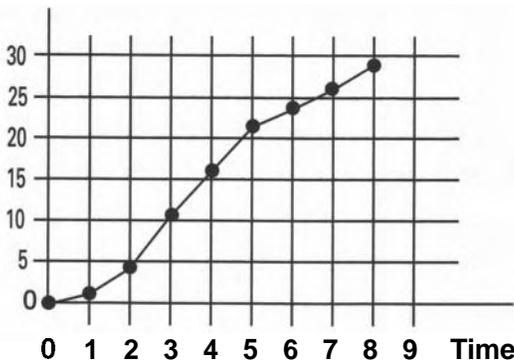
**Figure 8-4. Plot showing project behind schedule
and overspent.**

Date of
Analysis

Time
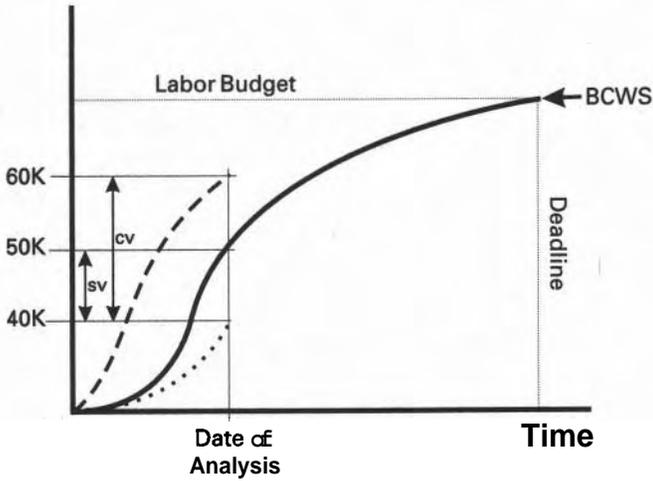
cv = **cost variance**
sv = **schedulevariance**

------ **ACWP**

.... **BCWP**

Figure 8-5. Project ahead of schedule, spending correctly.

Date of
Analysis

Time
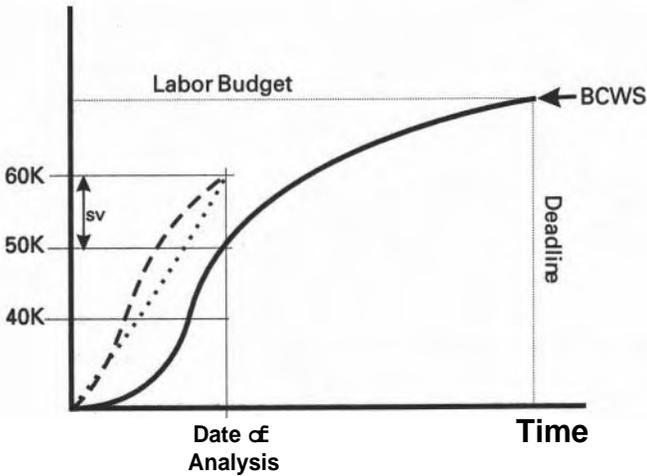
cv = **cost variance**
sv = **schedulevariance**

- - - - - **ACWP**

.... **BCWP**

The next set of curves illustrates another status. In Figure 8-6, the BCWP and the ACWP curves are both at 40K. This means the project is behind schedule and under budget. However, because the manager spent 40K and got 40K of value for it, spending is correct for what has been done. There is a *schedule variance,* but not a spending variance.

Figure 8-7 looks like Figure 8-4, except that the ACWP and the BCWP curves have been reversed. Now the project is ahead of schedule and underspent.

Variance Analysis Using Hours Only

In some organizations, project managers are held accountable not for costs but only for the hours actually worked on the project and for the work actually accomplished. In this case, the same analysis can be conducted by stripping the dollars off the figures. This results in the following:

■ BCWS becomes Total Planned (or Scheduled) Hours

► BCWP becomes Earned Hours (Scheduled hours x % work accomplished)

► ACWP becomes Actual Hours Worked

Using hours only, the formulas become:

$$\text{Schedule Variance} = \text{BCWP} - \text{BCWS} =$$
$$\text{Earned Hours} - \text{Planned Hours}$$

$$\text{Labor Variance} = \text{BCWP} - \text{ACWP} =$$
$$\text{Earned Hours} - \text{Actual Hours Worked}$$

Tracking hours-only does lead to one loss of sensitivity. ACWP is actually the composite of a labor rate variance times a labor-hours variance. When only labor-hours are tracked, you have no warning that labor rates might cause a project budget problem. Nevertheless, this method does simplify the analysis and presumably tracks the project manager only on what she can control.

Figure 8-6. Project is behind schedule but spending correctly.
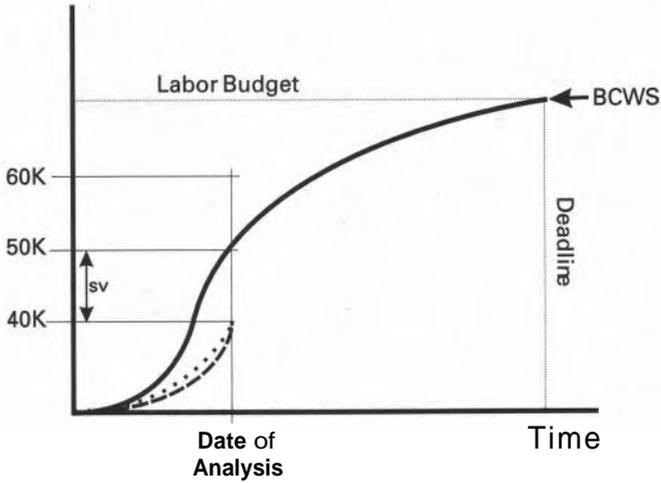
cv = cost variance   ----- ACWP
sv = schedulevariance   .... BCWP

Figure 8-7. Project is ahead of schedule and underspent.
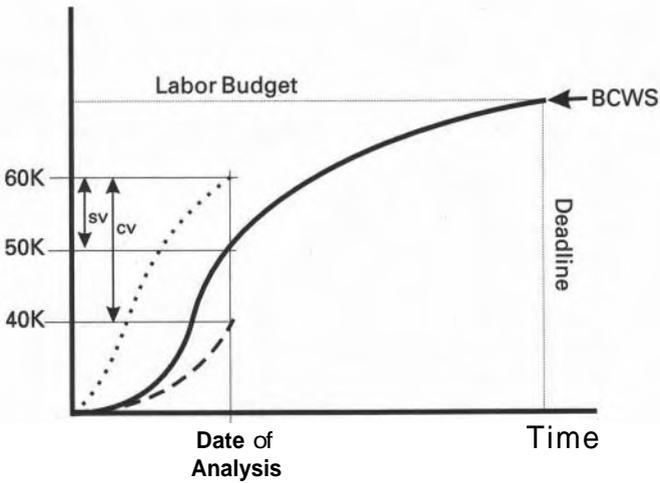
cv = cost variance   ----- ACWP
sv = schedule variance .... BCWP

Responding to Variances

It is not enough to simply detect a variance. The next step is to understand what it means and what caused it. Then you have to decide what to do to correct for the deviation. Earlier I explained that there are four responses that can be taken when there is a deviation from plan. Which of these you choose will depend in part on what caused the deviation. Following are some general guidelines:

▶ When ACWP and BCWP are almost equal and larger than BCWS (see Figure 8-5), it usually means that extra resources have been applied to the project, but at the labor rates originally anticipated. This can happen in several ways. Perhaps you planned for weather delays, but the weather has been good and you have gotten more work done during the analysis period than intended, but at the correct cost. Thus, you are ahead of schedule but spending correctly.

▶ When ACWP and BCWP are nearly equal and below BCWS (see Figure 8-6), it usually means the opposite of the previous situation; that is, you have not applied enough resources. Perhaps they were stolen from you, perhaps it has rained more than you expected, or perhaps everyone has decided to take a vacation at once. The problem with being in this position is that it usually results in an overspend when you try to catch up.

When ACWP is below BCWS and BCWP is above BCWS (see Figure 8-7), you are ahead of schedule and underspent. This generally happens because the original estimate was too conservative (probably padded for safety). Another possibility is that you had a lucky break. You thought the work would be harder than it was, so you were able to get ahead. Sometimes it happens because people were much more efficient than expected. The problem with this variance is that it ties up resources that could be used on other projects. The economists call this an *opportunity* cost. There is also a good chance that if you were consistently padding estimates and were bidding against other companies on projects, you probably lost

some bids. If your competitor is using average values for time esti-
mates while you are padding yours, then your figures are likely to
be higher, and you **will** lose the bid.

Acceptable Variances

What are acceptable variances? The only answer that can be
given to this question is, "It all depends." If you are doing a well-
defined construction job, the variances can be in the range of
± 3–5 percent. If the job is research and development, acceptable
variances increase generally to around ± 10–15 percent. When
the job is pure research, the sky is the limit. Imagine, for exam-
ple, that you worked for a pharmaceutical company and your
boss said, "Tell me how long it **will** take and how much it will
cost for you to discover and develop a cure for AIDS."

For every organization, you have to develop tolerances
through experience. Then you start trying to reduce them. *All*
progress is an attempt to reduce variation in what we do. We **will**
never reduce it to zero, without eliminating the process alto-
gether, but zero has to be the target.

Using Percentage Complete to
Measure Progress

The most common way to measure progress is to simply esti-
mate percentage complete. This is the BCWP measure, but
BCWP is expressed as a dollar value, whereas percentage com-
plete does not make that conversion.

When percentage complete measures are plotted over time,
you tend to get a curve like the one shown in Figure 8-8. It rises
more or less linearly up to about 80 or 90 percent, then turns
horizontal (meaning no further progress is being made). It stays
there for a while; then, all of a sudden, the work is completed.

The reason is that problems are often encountered near the
end of the task, and a lot of effort goes into trying to solve them.
During that time, no progress is made.

Figure 8-8. Percent complete curve.
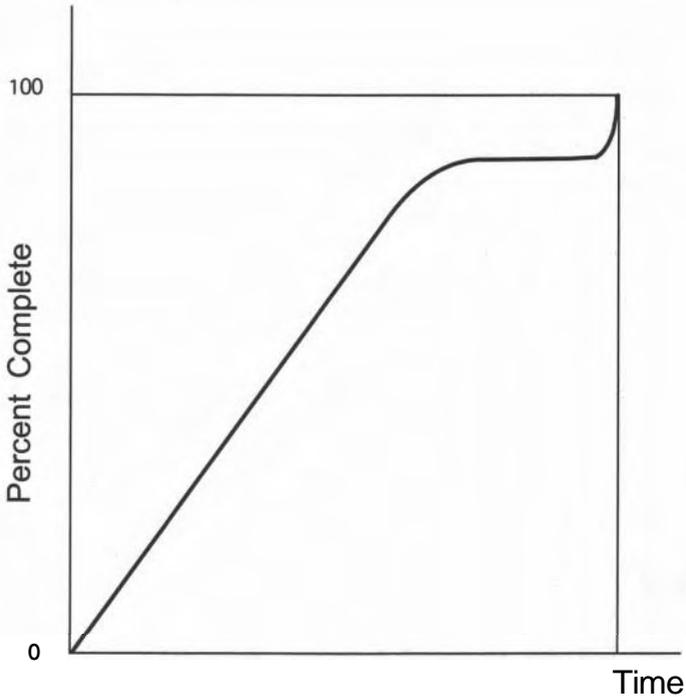
Another part of the problem is in knowing where you are to begin with. We have already said that you are generally estimating progress. Consider a task that has a ten-week duration. If you ask the person doing that task where he is at the end of the first week, he is likely to tell you, "10 percent"; at the end of week two, "20 percent"; and so on. What he is doing is making a reverse inference. It goes like this. "It is the end of the first week on a ten-week task, so I must be 10 percent complete." The truth is, he really doesn't know where he is. Naturally, under such conditions, control is very loose. Still, this is the only way progress can be measured in many cases.

■■
Key Points to Remember

Control is exercised by analyzing variances from the plan.

Well-defined projects can achieve tighter control over variations than poorly defined ones.

► There is a tendency to sacrifice quality when deadlines are difficult to meet.

It is not enough to recognize a variance. Its cause must **be** determined so that corrective action can be taken.

Acceptable variances can be determined only through experience. Every system has a capability. Your team may have the ability to maintain better tolerances on their work than another team.
■■

Exercise . . . .

Consider the report in Figure 8-9, showing earned value figures for a project. Answer the questions by analyzing the data. Answers are provided in the Answers section at the back of the book.

Figure 8-9. Earned value report.

	Cumulative-to-date			Variance		At Completion		
WBS #	BCWS	BCWP	ACWP	SCHED.	COST	BUDGET	L. EST.	VARIANCE
301	800	640	880	−160	−240	2,400	2,816	−416

Questions:

1. Is the task ahead or behind schedule? By how much?

2. Is the task overspent or underspent? By how much?

3. When the task is completed, will it be overspent or underspent?

Managing the Project Team

The previous chapters have concentrated primarily on the tools of project management—how to plan, schedule, and control the work. Unfortunately, far too many project managers see these tools as all they need to manage successfully. They assemble a team, give them their instructions, then sit back and watch the project self-destruct. Then they question whether there might be some flaw in the tools.

In all likelihood, the problem was with how people were managed. Even in those cases where a problem with the tools may have existed, it is often the failure of people to properly apply them that causes the problem, so, again, we are back to people.

The tools and techniques of project management are a *necessary* but not a *sufficient* condition for project success. As I have stated, if you can't handle people, you will have difficulty managing projects, especially when the people don't "belong" to you.

Related to this is the need to turn a project *group* into a team. Far too little attention is paid to team building in project management. This chapter offers some suggestions on how to go about it.

Team Building

Building an effective team begins on the first day of the team's existence. Failure to begin the team building process may result in a team that is more like a group than a team. In a group, members may be involved in but not committed to the activities of the majority.

Teams don't just happen—they must be built!

The problem of commitment is a major one for both organizations and project teams. It is especially significant in matrix organizations, in which members of the project team are actually members of functional groups and have their own bosses but report to the project manager on a "dotted-line" basis.

Later in this chapter, I present rules for how a project manager can develop commitment to a team. For now, let us turn to how to get a team organized so that it gets off to the right start. (For an in-depth treatment of this topic, see my book Team-Based Project Management.)

Promoting Teamwork Through Planning

A primary rule of planning is that those individuals who must implement the plan should participate in preparing it. Yet leaders often plan projects by themselves, then wonder why their team members seem to have no commitment to the plans.

All planning requires some estimating—how long a task will take, given the availability of certain resources, and so on. In my seminars, I ask participants, "Do you often find that your boss thinks you can do your work much faster than you actually can?" They laugh and agree. As I tell them, it seems to be some kind of psychological law that bosses are optimistic about how long it will take their staffs to get a job done.

When a manager gives a person an assignment that allows inadequate time to perform, the individual naturally feels discouraged, and her commitment is likely to suffer. She might say, "I'll give it my best shot," but her heart isn't really in it.

Getting Organized

Here are the four major steps in organizing a project team:

1. Decide what must be done, using work breakdown structures, problem definitions, and other planning tools.

2. Determine staffing requirements to accomplish the tasks identified in the first step.

3. Recruit members for the project team.

4. Complete your project plan through participation of team members.

Recruiting

Following are some of the criteria by which team members should be selected:

► The candidate possesses the skills necessary to perform the required work at the speed needed to meet deadlines.

■ The candidate will have his needs met through participation in the project (see the March and Simon rules discussed later in this chapter).

▄ The applicant has the temperament to fit in with other team members who have already been recruited and with the project manager and other key players.

► The person will not object to overtime requirements, tight timetables, or other project work requirements.

Clarifying the Team's Mission, Goals, and Objectives

Peters and Waterman have said that excellent organizations "stick to their knitting." They stick to what they are good at and do not go off on tangents, trying to do something they know nothing about. (Imagine, as an example, a hockey team deciding to play basketball.)

Numerous case studies and articles have been written about organizations that went off on tangents, at great cost, because they forgot their mission. The same can happen to project teams. If members are not clear on the team's mission, they will take the team where they think it is supposed to go, and that may not be the direction intended by the organization. The procedure for developing a mission statement is covered in chapter **3,** so no more will be said about it here. However, working with your team to develop a mission statement is a good team-building activity in itself.

> If possible, the entire team should participate in developing the team's mission statement. This is a tremendous team-building activity in itself!

Conflicts **Between** Individual Goals and the Team's Mission

Experience has shown that team members are most committed to a team when their individual needs are being met. Sometimes members have what are called *hidden* agendas, personal objectives that they do not want anyone to know about, because they are afraid other members **will** try to block them if their objectives are known. Since a manager should try to help individual members achieve their personal goals, while achieving, team goals as well, the team leader needs to bring hidden agendas into the open so that the individual can be helped to achieve his goal. Of course, a person may occasionally have a goal that runs so counter to the team's goals that no reconciliation is possible. In that case, if the team leader can discover what the person's goal is, the individual can (ideally) be moved to another team in which his goal can be reached.

> A manager should try to satisfy the needs of the organization, while simultaneously helping individuals satisfy their own needs through participation in the project.

Team Issues

There are four general issues with which a team must deal. These are goals, roles and responsibilities, procedures, and relationships. In this chapter, we have dealt with clarifying the team's mission, goals, and objectives. This is always the first and most important step in developing a team.

Every team must deal-with: goals roles & *responsibilities* **procedures** *relationships*

Once that is done, people must understand their roles. These must be clearly defined. What is expected of each individual, and by when? The one problem that seems common is that team leaders think they clearly communicate this information to team members. Yet, when you ask team members if they are clear on their goals and roles, you frequently get a negative response.

The problem is with our failure to solicit feedback from team members to be really sure that they understood; in addition, members themselves are sometimes reluctant to admit that they haven't understood. This appears to be a result of our tendency in school to put people down for asking "stupid questions." So, rather than admit that they don't understand, they interpret what they have been told and try to do the job the best they can.

Project leaders must establish a climate of open communication with the team, in which no one feels intimidated about speaking up. The best way to do this is to comment on the problem. "I know some of you may feel reluctant to speak up and say you don't understand, but we can't operate that way. Please feel free to be candid. If you don't understand, say so. If you don't agree with something, say so.

There is no such thing as a stupid question—except perhaps the one you were afraid to ask.

That is the only way we can succeed. We will be lucky to have time to do the job once, much less find time to

do it over because one of you failed to understand what was expected."

I have also found that people respond very positively when I am willing to admit that I don't understand something myself or am apprehensive or concerned about a project issue. If you project an air of infallibility, no one else is likely to admit a weakness. But, then, who wants to deal with a demigod? A little human frailty goes a long way toward breaking down barriers. I know this contradicts what some managers have been taught. The macho notion of infallibility has been with us for a long time, and I believe it is the cause of many of our organizational problems. It is time to abandon it for reality.

Working Out Procedures

Dealing with *how we do it* comes next. The key word here is *processes.* The work must be done as efficiently and as effectively as possible, and improvement of work processes is a very important issue today. It is commonly called *re-engineering* and is the analysis and improvement of work processes to make the organization more competitive.

The difficulty that most teams have with process is that they get so focused on doing the work that they forget to examine how it is done. Periodically, a team should stop working long enough to examine its processes and to see whether it could use better approaches. Otherwise, the team might get very good at doing the work badly.

> So-called personality conflicts are often simply the result of people's lack of good interpersonal skills. This lack can be resolved through training.

Relationships in Teams

Friction occurs in nearly every interaction between human beings. There will be misunderstandings, conflicts, personality clashes, and petty jealousies. Project managers must be prepared to deal with these. In fact, if you really dislike having to deal with the behavioral prob-

lems that occur on projects, you should ask yourself whether you really want to manage projects at all. Like it or not, the behavioral problems come with the job, and failure to deal with them **will** sink a project eventually.

One thing to be aware of is that many personality clashes are the result of people's lack of good interpersonal skills. We have never been taught how to sit down and work out differences with others, so when the inevitable conflict happens, the situation just blows up. The best way to minimize the impact of such problems is to provide training for all team members (including yourself) in interpersonal skills. This area has been sorely neglected in many organizations because there seems to be no bottom-line impact. It is hard to demonstrate that there **will** be a $10 return on a $1 training investment.

Because of our inability to quantify the benefits of skills training, we don't provide it. Yet, if we have capital resources that don't work well, we spend whatever is necessary to correct the problem. Interestingly, our human resources are the only ones that are renewable almost indefinitely, but we fail to take steps to keep them functioning effectively. **As** a project manager, you owe it to yourself to manage this aspect of the job.

Stages in a Team's Development

There are a number of models that describe the stages that teams or groups go through on the way to maturity. One of the more popular ones has self-explanatory titles for the stages: forming, storming, norming, and performing.

In the forming stage, people are concerned with how they **will** fit in and with who calls the shots, makes decisions, and so on. During this stage they look to the leader (or someone else) to give them some structure—that is, to give them a sense of direction and to help them get started. **A** leader's failure to do this may result in loss of the team to some member who exercises what we call informal leadership.

The storming stage is frustrating for most people. When the team reaches this stage, people begin to question their goals. Are

they on the right track? Is the leader really leading them? They sometimes play *shoot the leader* during this stage.

At the *norming* stage they are begin-
ning to resolve their conflicts and settle
down to work. They have developed
norms (unwritten rules) about how they
will work together, and they feel more
comfortable with one another. Each indi-
vidual has found her place in the team and
knows what to expect of the others.

Finally, when the team reaches the *per-
forming* stage, the leader's job is easier.
Members generally work well together
now, enjoy doing so, and tend to produce
high-quality results. In other words, we can
really call them a team at this point.

> The most **popular**
> terms for the
> **stages of team**
> **development are:**
> *forming*
> *storming*
> *norming*
> *performing*

Leading a Team Through the Stages

A newly formed team needs considerable structure, or it will not be able to get started. As I noted in the previous section, a leader who fails to provide such structure during stage 1, the *forming* stage, may be rejected by the group, which will then look for leadership from someone else. A *directive* style of leadership is called for in the forming stage.

During this stage, members also want
to get to know one another and want to
understand the role each member will
play on the team. In stage 1, the leader
must help team members get to know one
anther and help them become clear on
goals, roles, and responsibilities. Leaders
who are very task oriented tend to make a

> A *directive* **style of**
> **leadership is called**
> for when a team is in
> **the *forming* stage.**

major error here: they just tell the team to "get to work," with-
out helping members get to know one another. They view such
purely "social" activities as a waste of time; surely members can
attend to such things themselves. Although it seems obvious, it

is hard to see yourself as a team when you don't know some of the "players."

Getting the team started with a kickoff party or dinner is one way to let members become acquainted in a purely social way, with no pressure to perform actual task work. If this is not feasible, there must be some mechanism for letting people get to know each other.

As the group continues to develop, it enters stage 2, *storming*. Here, people are beginning to have some anxiety. They start to question the group's goal: are we doing what we're supposed to be doing? The leader must use *influence* or *persuasion* to assure them that they are indeed on track. They need a lot of psychological support, as well. They must be assured by the leader that they are valued, that they are vital to the success of the team, and so on. In other words, members need some stroking in this stage.

A selling or influence style of leadership is appropriate at the storming stage.

There is a tendency to try to skip this stage, as managers feel uncomfortable with the conflict that occurs. To sweep such conflict under the rug and pretend that it doesn't exist is a mistake. The conflict must be *managed* so that it does not become destructive, but it must not be avoided. If it is, the group will keep coming back to this stage to try to resolve the conflict, and this will inhibit progress. Better to pay now and get it over with.

As the team enters stage **3,** *norming,* it is becoming closer knit. Members are beginning to see themselves as a team and take some sense of personal identity from membership in the group.

In the norming stage, the leader should adopt a participative style of leadership.

Members are now involved in the work, are becoming supportive of each other, and, because of their cooperation, can be said to be more of a team than a group at this point. The leader

needs to adopt a participative style in this stage and share decision making more than in stages 1 and 2.

By the time a group reaches stage 4, *performing,* it is a real team. The leader can generally sit back and concentrate on what-if analysis of team progress, planning for future work, and so on. This is a delegative style of leadership and is very appropriate. The team is achieving results, and members are usually taking pride in their accomplishments. In this stage, there should be signs of camaraderie, joking around, and real enjoyment in working together.

> *Delegative* **leader-ship is the proper** *style in the per-forming* **stage** *of* **a team's development.** Note that delega-tive *does not mean* | *abdication!*

It is important to remember that no team stays in a single stage forever. If it encounters obstacles, it may drop back to stage **3,** and the leader can no longer be delegative but must back up to the stage **3** management style, which is participative.

Membership in project teams often changes. When new members come on board, you should consider that for a short time the team will fall back to stage 1, and you will have to take it back through the stages until it reaches maturity again. It is especially important that you help everyone get to know the new member and what his role will be in the team. This does take some time, but it is essential if you want the team to progress properly.

Developing Commitment to a Team

At the beginning of this chapter, I pointed out that developing commitment to a project team is a major problem for project managers. Team members are often assigned to a project simply because they are the best available people, not because they are the best people for the job. When this happens, they may have no commitment to the team.

In their book *Organizations* (1958), March and Simon present five rules for developing commitment to a team or organization. Those rules are listed below:

RULES FOR DEVELOPING COMMITMENT TO
A PROJECT TEAM

1. Have team members interact frequently so that they gain a sense of being a team.

2. Be sure that individual needs are being met through participation in the team.

3. Let all members know why the project is important. People don't like working on a "loser."

4. Make sure all members share the goals of the team. One bad apple can spoil the barrel.

5. Keep competition within the team to a minimum. Competition and cooperation are opposites. Let members compete with people outside the team, not within it.

Note that the first rule cannot always be followed if the team is scattered geographically. In that case, members should "meet" frequently through teleconferencing. It is almost impossible to think of yourself as part of a team when the team never gets together in some manner.

Watch the movie ***Stand and Deliver*** *for an excellent example of true leadership.*

A Final Suggestion

If you want some good models of how to work with teams, take a look at the best coaches and see how they do it. Be careful, though, not to model the supermacho coach's behavior. That might work okay with a sports team, where people are there

because they want to be there, but it is unlikely to work well with a project team where the members are there because they *have* to be. I also suggest that you watch the movie *Stand and Deliver* and see how Jaime Escalante deals with his kids. Then, the next time you are tempted to complain that you have a lot of responsibility and no authority, ask yourself how a teacher (who has even less authority than you do) can get a bunch of kids to work so hard. How did he get them to go to summer school or take math two periods a day? Then you **will** begin to realize what true leadership is all about.

● ●
Key Points to Remember

Teams don't just happen—they must be built!

Having the entire team participate in planning is one way to start the team-building process.

Deal with goals, roles and responsibilities, procedures, and relationships, in that order.

▶ So-called personality conflicts are often caused by team members' poor interpersonal skills. For teams to function well, all members should receive training in this area.

▶ The style of leadership appropriate for a team depends on its stage of development. In the forming stage, it is directive. In storming, it is influencing. At the norming stage, switch to a participative style. Finally, when the team reaches the performing stage, you can be delegative.
● ●

How to Make Project Management Work in Your Company

t is one thing to know how to manage projects. It is another to get people to actually *do* it. Running by the seat of the pants *seems* a lot easier than doing all the planning, scheduling, and monitoring that have been presented in this book. Even when people invest three or four days in project management seminars, you find that they soon forget what they have been taught and go back to the old ways.

I have struggled with this problem for twenty years and finally have some answers. Here are suggestions on how to make the principles of project management work in your company:

Dr. W. Edwards Deming learned more than fifty years ago that if you don't get top management involved in a program, the program will be short-lived. This doesn't mean just having them pay lip service to it. As Tom Peters suggests in his book *Thriving on Chaos,* if an executive wants something to happen in the company, she has to change her calendar; she must spend time talking about project management, sit in on project planning or

review meetings, start asking to see people's project notebooks, and ask questions about how projects are doing. In other words, she must show an interest in the subject.

Companies must build into performance appraisals items that evaluate a project manager's use of the tools. They should reward people for practicing the best methods and, if necessary, sanction them when they do not. But be careful. Be sure upper management is not keeping managers from practicing good methodology.

It helps to have the entire team trained in the basics. After all, when you tell members of your team that you want them to do a WBS for their part of the project and they've never even heard the term before, they can't very well deliver. I have found that project managers generally need a minimum of three or four days' training in project management, and team members need about two days' training.

I have found that senior management should have a brief overview of the principles so that it knows what it is realistic to expect. One of the most common causes of project failures is unrealistic expectations on the part of senior managers. However, I have found that most senior managers are so busy that you can get them together for only about three hours—if you can even do that. We have finally videotaped a briefing and cut it down to one hour and fifteen minutes, just enough time for busy executives to learn what they need to know to support and drive the effort.

After the training is complete, pick a project that already has a pretty high probability of success—don't pick your hardest job; the probability of failure is too high—and have your trainer/consultant walk the team through the steps. This is the hand-holding phase, and I have found it to be essential (as have a number of major companies with which I have worked). It really helps to have someone assist the team in practicing what it has learned. All new procedures feel awkward when you first try

them, and an outside expert makes things go smoother. In addition, an outsider can be more objective than members of the team.

Plan small wins for people. Forget the Pareto principle. It's wrong in this particular instance, even from an economic point of view. According to Pareto, you should begin with your most important problems and solve them, then move on to the simpler ones. Sounds like good economic sense, but it isn't. It ignores the fact that the biggest problem is also likely to be the hardest to tackle, so people are more likely to fail, become demoralized, and give up. No sports team ranked number 10 would want to play the top-ranked team for its first game. It would prefer to play the ninth-ranked team maybe, or even the eleventh. Don't set the team up to be slaughtered!

Practice a lot of MBWA (management by walking around) as the project progresses, but do it to be helpful, not in the blame-and-punishment mode. Give people strokes for letting you know about problems early, not after they have turned into disasters. Don't be too quick to help them, though. Give them time to solve the problems themselves. Just ask them to keep you informed and tell them to let you know if they need help. Be a resource, not a policeman.

Do process reviews to learn and to try to improve whenever possible.

If you find you have a problem child on your team, deal with that person as soon as possible. If you don't know how to handle the problem, talk to someone who has the experience and who can help you. Don't ignore the problem, as it can wreck your entire team.

▶ Be very proactive, not reactive. Take the lead. Break roadblocks for your team members. Go to bat for them.

Have team members make presentations to senior management on their part of the job. Give them credit for their contributions. Build ownership.

If you are running a project where people are assigned temporarily but still report to their own bosses (the matrix organization), keep their managers informed about what they are doing. Try to build good relations with those managers. You may need their support to get the job done.

For those tasks on the critical path of the project, you may find that you have to physically collocate the people doing those activities so that you don't have them constantly pulled off to do other jobs. Major corporations are using this method more and more today on highly important projects.

It may be useful to consider setting up a *project support person or office* to do all scheduling for your project managers. Rather than have everyone try to master the software, it may be better to train one or two people to competence level, with users trained only enough to know the capability of the software. Under this scenario, project managers give raw data to the support group, which enters them into the computer and then gives back the schedule; the schedule is then massaged until it works. Subsequently, the support group does all updates, what-if analyses, and so on for the project manager.

Along this line, have a person assigned as *project administrator.* This person either does the project support or delegates it. He also sits in on project review meetings, holds the team's hands to walk members through planning and audits, and so forth. Naturally, you need to be running quite a few projects (at least ten to twenty) to justify this position. Such a person can be helpful when the people who are managing projects have little experience with managing or perhaps have poor skills in dealing with people, or both.

Benchmark other companies to find out what they do with project management. Note that, when they don't practice good methodology, that does not give you grounds for abandoning it yourself. I know of one major corporation that does not track actual work put into a project, yet the company is extremely successful. However, the fact that it doesn't track

work is going to lead to problems eventually. It does a lot of other things really well, however, and I would not hesitate to benchmark those things.

Have individuals take responsibility for being *champions* of various parts of the project management process. Perhaps you can make one person the earned value champion, who goes around the company trying to get everyone on board so that they all use that method. Another could take responsibility for dealing with WBS notation, and so on.

Join the Project Management Institute, attend its chapter meetings, and learn more about project management from other professionals.

Try to read current management books and glean everything you can from them that will help you do your job better. Managing projects is a demanding job, and you need all the help you can get.

Consider changing the structure of the organization to be project based. Tell all functional managers that they exist to serve the needs of projects. Many of them **will** scream. Some may even quit. But, in today's world, where most of what gets done in organizations is in project format, this makes good sense.

Set up a project management function, with dedicated project managers. You don't have everyone doing accounting. Not everyone is good at it. This is also true of project management. By making it a function, like all the others, you provide a way for dedicated individuals to hone their skills and get really good at the job. An excellent resource for this is Graham & Englund, *Creating an Environment for Successful Projects (1997)*.

Look at managing projects as a *challenge* or even as a *game*. If it doesn't strike you that way, it probably won't be very exciting. Experiment with new approaches. Find out what works and keep it. Throw out what does not.

Finally—good luck!

Answers to
Chapter Questions

Chapter 1

1. c
2. d
3. a
4. b

Chapter 2

You should decide on project strategy before you begin implementation planning. At that point, you should develop tactics to execute strategy and plan logistics so that people will have what they need to execute the tactics.

Chapter 4

WBS for camping trip:

Figure A-1. WBS for camping trip.
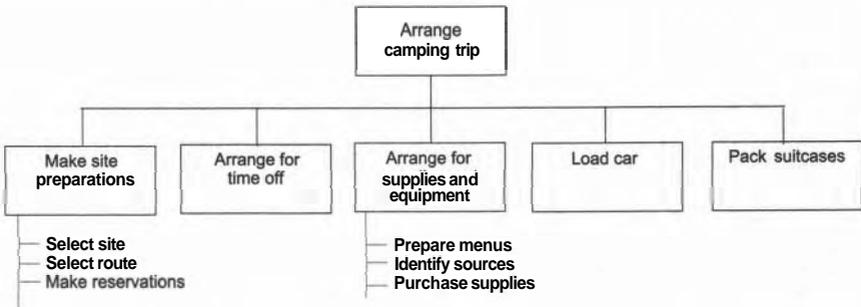
Chapter 5

Solution to the WBS exercise:

Figure A-2. Solution to WBS exercise.
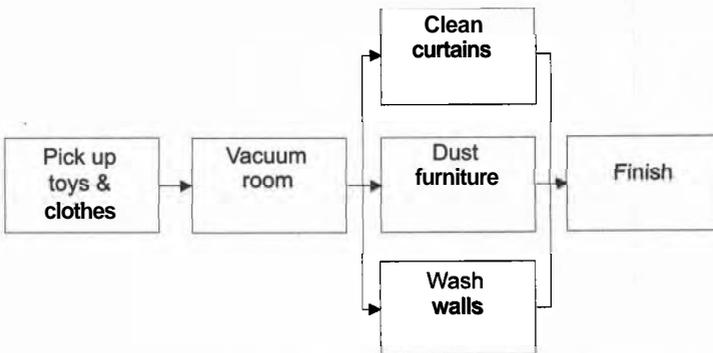
Chapter 6

Solution to the scheduling exercise:

Figure A-3. Solution to scheduling exercise.
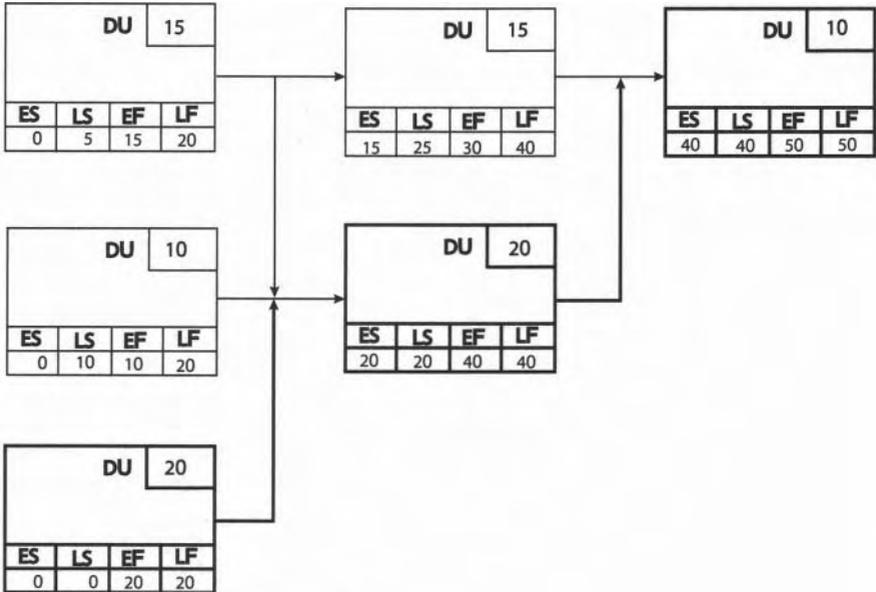
Chapter 8

1. It is behind schedule **by** $160 worth of work.

2. It is overspent **by** $240.

3. It **will** be overspent **by** $416.

References and Reading List

Argyris, Chris. *Overcoming Organizational Defenses: Facilitating Organizational Learning.* Boston: Allyn and Bacon, 1990.

Deming, W. Edwards. *Out of the Crisis.* Cambridge: Massachusetts Institute of Technology, 1986.

Graham, Robert J., and Randall L. Englund. *Creating an Environment for Successful Projects.* San Francisco: Jossey-Bass, 1997.

Lewis, James. *Team-Based Project Management.* New York: AMACOM, 1997.

――――. *Mastering Project Management.* New York: McGraw-Hill, 1998.

――――. *The Project Manager's Desk Reference, 2d* ed. New York: McGraw-Hill, 2000.

――――. *Project Planning, Scheduling, and Control,* 3d ed. New York: McGraw-Hill, 2001.

March, James, and Herbert Simon. *Organizations.* New York: Wiley, 1958.

Packard, Vance. *The Pyramid Climbers.* New York: McGraw-Hill, 1962.

Peters, Tom. *Thriving on Chaos.* New York: Knopf, 1987.

Peters, Tom, and Bob Waterman. *In Search of Excellence.* New York: Warner,1984.

Sabbagh, Karl. *21ˢᵗ Century Jet.* New York: Scribner, 1996.

Wysocki, Robert K, and James P. Lewis. *The World-Class Project Manager.* Boston: Perseus, *2000.*

Index

About Jim Lewis

James P. Lewis, Ph.D. is an experienced project manager who now teaches seminars on the subject throughout the United States, England, and the Far East. His solid, no-nonsense approach is largely the result of the fifteen years he spent in industry, working as an electrical engineer engaged in the design and development of communication equipment. He held various positions, including project manager, product engineering manager, and chief engineer, for Aerotron, Inc., and ITT Telecommunications, both of Raleigh, North Carolina. He also was a quality manager for ITT Telecom, managing a department of sixty-three quality engineers, line inspectors, and test technicians.

While he was an engineering manager, he began working on a doctorate in organizational psychology, because of his conviction that a manager can only succeed by developing good interpersonal skills.

Since 1980, Dr. Lewis has trained over 20,000 supervisors and managers throughout the United States, Argentina, Canada, Germany, India, Indonesia, Malaysia, Mexico, Singapore, Sweden, Thailand, and England. He has written articles for *Training and Development Journal, Apparel Industry Magazine,* and

Transportation and Distribution Magazine and is the author of *Project Planning, Scheduling, and Control,* 3d ed.; *Mastering Project Management;* and *The Project Manager's Desk Reference,* 2d ed.; published by McGraw-Hill, and *Fundamentals of Project Management; How to Build and Manage a Winning Project Team;* and *Team-Based Project Management;* published by AMACOM, a division of the American Management Association. He is co-author, with Bob Wysocki, of *The World-Class Project Manager,* published by Perseus in 2001. The first edition of *Project Planning, Scheduling, and Control* has been published in a Spanish edition, and the AMACOM book *Fundamentals of Project Management* has been published in Spanish and Portuguese.

Dr. Lewis has a B.S. in Electrical Engineering and a Ph.D. in Psychology, both from North Carolina State University in Raleigh. He is a member of several professional societies, including the Project Management Institute and The American Society for Training and Development. He is president of The Lewis Institute, Inc., a training and consulting company specializing in project management, which he founded in 1981.

He is married to the former Lea Ann McDowell, and they live in Vinton, Virginia, in the Blue Ridge Mountains.