15

Optimizing Resources in a Multiproject Environment

This chapter's goal is to develop the remaining tools and concepts necessary to create a comprehensive resource management doctrine. This enables you to optimize resources in a multiproject environment.

The previous chapter developed the concept of the infinite resource pool, which enables you to outsource when necessary to achieve your strategic goals while capturing opportunistic and creative projects. It also helps you develop portfolio agility, making your portfolio more flexible in a dynamic environment.

This chapter builds on those themes to fine-tune your organization, squeezing more production from the system without impinging on morale, quality, or portfolio agility. This chapter uses a systems approach to resource management. This enables us to examine project management in a real work environment, recognizing that multiple projects occur simultaneously, that functional managers and project managers must exchange resources so that they can both accomplish their goals, and that both must remain focused on the organization's vision and strategy. First, I present the final tools needed to expand project and portfolio agility to organizational agility.

Project Management and Organizational Agility

Globalization and the availability of information have inspired a new concept: **organizational agility.** As new markets open, new technologies develop, and new geopolitical situations arise, organizations must move quickly to maintain their market positions.

In the previous chapter, I introduced the concept of **portfolio agility:** the capability to adjust the portfolio quickly to capture creative or opportunistic projects without affecting the rest of the portfolio. These tools enable us to grow the concept of agility to the organization as a whole.

However, organizational agility introduces a conundrum. All organizations must run efficiently to survive. Efficiency comes from consistency. Agility inherently requires extra resources. As situations arise, the organization must move in the new direction. This can

occur in only one of two ways: The organization abandons its current activities, or the organization has sufficient resources to move toward the new activities without affecting existing activities.

The right tools and a systems approach enable you to strike a balance between agility and efficiency.

We've already addressed one major tool for improving organizational throughput: the infinite resource pool. To this, we add the following:

- 100 percent utilization fallacy
- Multitasking and single-tasking
- Cross-training
- Theory of Constraints
- Synchronizers and throttlers
- Deliverables-oriented assignments

The 100 Percent Utilization Fallacy

You walk into a fast-food restaurant. Let's assume that you are the only customer in the restaurant. Three smiling faces stand behind the counter ready to take your order. You walk up to the counter, place your order, pay for and receive your food. Let's assume that the entire transaction lasts one minute. You are happy because you were served immediately—this is, after all, a fast-food restaurant.

Now consider that you enter the restaurant with two friends. Again, you're served immediately because three smiling faces are behind the counter; you and your friends can place your orders simultaneously.

Now consider that you enter the restaurant with three friends. Four customers and three smiling faces mean that one of you must wait for one minute while the other three place their orders.

In the first two scenarios, you were served immediately because at least one person behind the counter was idle. In the third scenario, one of you had to wait because all the servers were busy.

When we translate this fast-food restaurant analogy to project management, we associate the customers entering the restaurant to work packages ready to be started. The smiling faces behind the counter are resources capable of working on those work packages. When a work package is ready to start, one of two conditions must exist:

1. The resources to do the work are idle, so the work can start immediately.
2. The resources to do the work are not idle, so the resource manager makes a choice:
   a. The resources stop what they're doing and start the work package.
   b. The resources continue what they're doing and delay the work package.

Figure 15.1 illustrates this decision path.

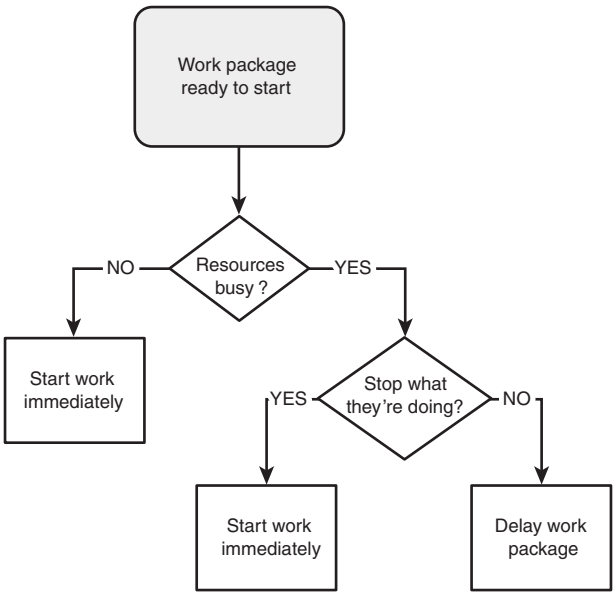

Figure 15.1 Resource utilization decision flow

When a project manager is ready to start a work package, the resources must be idle or we must delay either the work package or the resources' current activity. This decision might be easy for a single event; however, because most organizations do more than one project and work packages might require multiple resources, schedule demands tend to pressure managers into hoarding resources to ensure their survival. The result is chaos and anarchy—project managers and functional manager fighting over resources.

The branch of science that handles such situations is **queuing theory**, a concept that every fast-food restaurateur knows well. When your order-takers are busy, the next customer coming in the door must wait in a queue (Americans call it a line). Queuing theory contends that, for any system, the probability of being placed in a queue is proportional to how busy the server is. Mathematically:

Probability of Waiting = Percent Resource Utilization

This means that when your resources are 100 percent utilized, every work package will be delayed, and you can guarantee that every project will be late. The way to handle this is to leave resources idle. This leads to a conundrum:

Idle resources cost money, but so do late projects.

Fortunately, project management offers us one aid: the concept of critical path. Recall from Chapter 3, "Project Definition and Planning," that the critical path tells us which work is time critical. All project work isn't time critical; we can delay some work for some amount of time without affecting the project (the amount of time you can delay a task without affecting the project's end date is called **float**). If your project managers don't determine the critical path and floats, they must assume that all activities are schedule critical, which exacerbates the problem. Therefore, we realize this point:

Multiproject environments cannot run efficiently without the use of critical path concepts.

Some project activities must start as soon as they are ready, or they will delay the project. Other activities can wait a short amount of time without impact, enabling the resources that are assigned to those activities to first finish their current work.

One more aid helps improve efficiency while keeping projects on track. To examine this aid, let's go back to our fast-food restaurant analogy. Only two ways exist to ensure that you service the customer immediately:

1. Have your order-takers idle.
2. Have them doing something they can drop at a moment's notice. (Let's call these **background activities**.)

For example, the order-takers shouldn't mop a floor. If a bus pulls up to the restaurant, they would need to leave a wet floor to serve the customers, which is dangerous. However, they could stack boxes for inventory if the inventory was not perishable (it didn't need to get into the freezer quickly) and didn't impose a safety hazard.

We now have a framework to improve resource utilization without affecting project schedules. If a work package is on the critical path, you can assign its resources to other tasks as long as they can drop those tasks at a moment's notice without negative effects. Similarly, if a task has float, you can assign its resources to other tasks as long as they can finish those tasks before the work package's float expires. This situation enables you to maintain efficiency while ensuring that your projects complete on time. However, this situation presents one more conundrum: losing efficiency while multitasking.

Multitasking and Single-Tasking

In Chapter 2, "Project Management Framework and Structure," I described two kinds of work estimates: effort and duration. Effort estimates establish the actual amount of dedicated time a human resource expends on an activity. It was previously called **man-hours** or **man-days,** and is now called **full-time equivalents (FTEs),** for political correctness. Duration estimates determine how long a task takes on a calendar. So a construction project manager asks an electrician how long it takes to wire a new house. The electrician offers two answers: The effort estimate is five days, but it will take one month on a calendar. The difference is that the electrician must wait for the inspectors to inspect the rough electrical, then they must wait for the drywall and paint, and then the electricians can come back and hook up the plugs and put on the covers.

For this discussion, I focus on effort estimates. A project manager asks a team member to provide an effort estimate for a work package. The team member says, "Boss, if it's just me doing the work, I need 80 hours of effort."

Now let's give this individual some help by adding more people to the work package. Adding people increases the amount of communication, coordination, and conflict—which increases the effort required to do the activity.

Now let's take the other perspective. If we multitask individuals, we frequently pull them off the work package to do other work. When they're done with the other work and return to our work package, they need to reorganize, remember where they were, set up their workspace, and so on. This also increases the amount of effort required to do the work package. Figure 15.2 illustrates this effect.

This leaves us with one obvious conclusion:

The most efficient way to manage human resources is to give them a task and leave them alone.

In Chapter 1, "The Goals of Project Management," I introduced the "Warm-Body Syndrome": the tendency for managers to grab any warm body to perform some task that they want done. This practice can only increase the amount of both effort and duration required to complete an activity. While debunking this practice in seminars, I frequently face the argument that "We don't have time to single-task people." My response: "Interesting. What you're telling me is that you don't have time to save time."

In practice, some people are good multitaskers and others are not. Engineers, computer programmers, scientists, and other high-technology disciplines tend to demand single-tasking individuals. The reason is that single-tasking enables an individual to drill down deep into the discipline or science. If you're multitasking, you don't have time to drill down deep into the science; you only have time to skim the surface. Consider Figure 15.3.

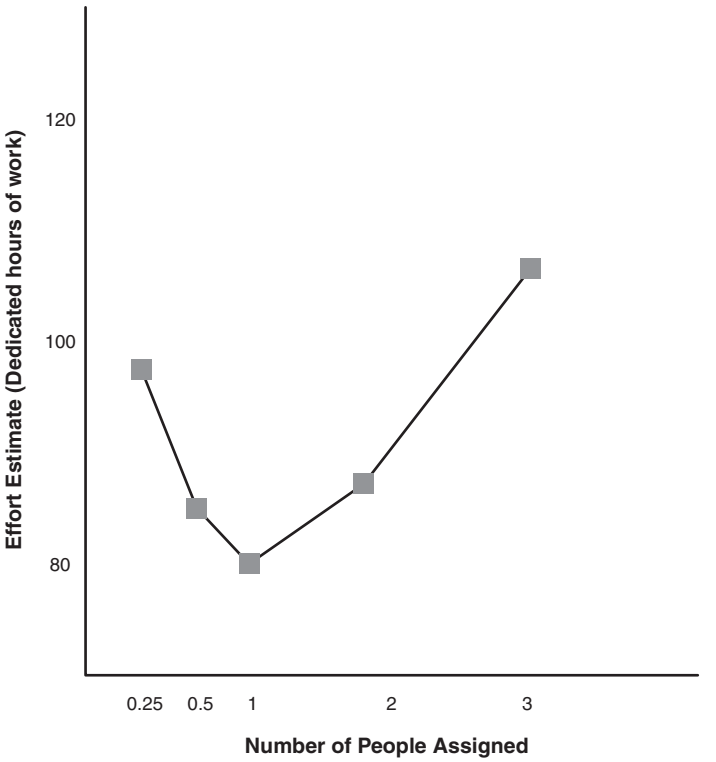

Figure 15.2 Work estimate effects of multitasking

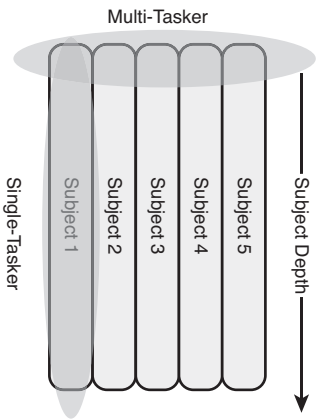

Figure 15.3 Single-tasking and multitasking

The multitasker can cover multiple subjects but can only skim the surface. The single-tasker specializes in one subject and understands it well.

Project and process work demands both kinds of people: single-taskers and multitaskers.

However, multitaskers are less accurate when they become overwhelmed, and even single-taskers need occasional distractions. Don't overwhelm multitaskers—give them what they can take.

You can give single-taskers productive distractions by providing them with background activities. These background activities are the same as those I described in the fast-food restaurant example. In a project world, these activities might include reviewing lessons learned from other projects, participating in PMO activities that won't be due for a long time, participating in peer reviews, taking seminars or participating in online training, and others. While this concept may appear to contradict my earlier statements, these background activities aren't time critical and their purpose is to make the task performer more efficient, not less efficient.

Cross-Training

Cross-training is more than a necessary element to develop a successful multiproject environment and organizational agility; it also offers many ancillary benefits to the organization as a whole.

Cross-training is one of the simplest ways to improve culture, staff efficiency, and organizational agility.

An organization should cross-train for two reasons. First, cross-training helps your staff and managers understand the roles they play in the organization. This helps break down silos, improve workflow, and increase morale. Second, cross-training enables people to step into other roles when needed. Both benefits require little effort and produce a significant benefit (they have a high return on investment—ROI), and both are necessary to achieve organizational agility.

The approach for developing a sound cross-training strategy is similar for both objectives; they differ only in the level of detail. Use your organization's process maps (see Chapter 7, "Process Alignment") as your primary aid. For each resource that performs a process step, cross-train that resource on the input processes (the processes that create the inputs) and the output processes (the processes that use the outputs).

To illustrate, I present a high-level example designed to achieve our first objective. Consider an IT business analyst function. For our example, the business analyst's role is to develop

requirements for IT projects. The analyst for our example is developing requirements for a new accounts payable system for the finance department. The analyst accepts general needs from the finance department, writes requirements, and gives those requirements to both the systems architects who will design the systems and the quality control department that will test the system. Figure 15.4 illustrates the business analyst's workflow.

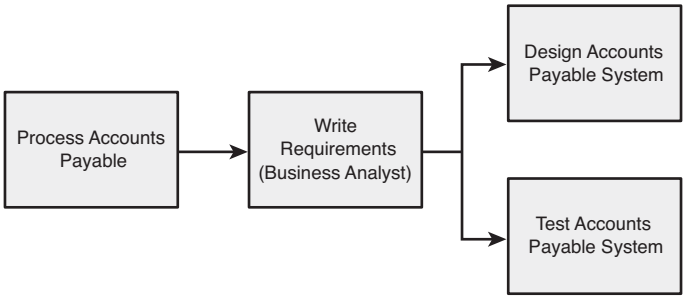

Figure 15.4 IT business analyst workflow

For this example, the business analyst should receive training in processing payables, systems design, and systems testing techniques. This training doesn't need to be extensive or thorough; conceptual training will improve the analyst's ability to do his or her job and interface with the finance, systems design, and quality control departments.

This example shows cross-training at a high level, which achieves our first goal. We wouldn't necessarily expect the business analyst to step into the systems design role, because that role is highly specialized. However, let's consider the converse view. We would also cross-train the systems designers on how to write requirements. With sufficient experience and training, I could expect a systems architect who possesses sufficient interpersonal skills and financial knowledge to step into the business analyst function when needed, achieving our second objective. Cross-training is both simple and inexpensive to implement and produces many organizational benefits. For our purposes in this book, cross-training provides us with the following:

- More efficient workflow
  - Shortened project durations
  - Improved quality
- Expanded resource pool
  - Staff capable of performing more functions as resource demands shift

The Theory of Constraints

Consider the simple production line shown in Figure 15.5. The line consists of three stages: A, B, and C. For this example, the production line makes the ubiquitous widget. Raw materials enter at Stage A. Stage A creates partially completed components (called **work in progress** in manufacturing). These components feed into Stage B. The output of Stage B feeds into Stage C, which finishes the widget.

Let's further assume that Stage A can produce the components for the equivalent of 500 widgets per hour. Stage B can produce 100 widgets per hour and Stage C can produce 250 widgets per hour. The overall throughput of our system is 100 widgets per hour, constrained by Stage B.

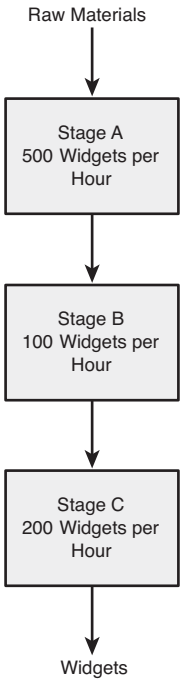

Figure 15.5 Throughput of a simple production line

In 1984, Dr. Eliyahu M. Goldratt published a book called *The Goal* that describes this simple concept. The book takes a systems approach to business processes to improve overall organizational effectiveness; I believe that any serious business person or organizational leader should read that book. The basis for the book is what Dr. Goldratt calls the theory of constraints (TOC). Dr. Goldratt postulates two concepts:[1]

1. Every system has at least one constraint.
2. At a specific point in time, almost all systems have only one constraint.

The first postulate is obvious because we realize that a system with no constrains could do anything with nothing instantly. The second postulate gives the TOC its strength. This means that if we want to improve a system, we need to find the constraint and either reduce its impact on the system or eliminate it. That action enables the system to make a quantum improvement until it hits the next constraint.

In our example, Stage B is the constraint. To improve the throughput of the widget production line, we need to improve the throughput of Stage B. Improving Stage A or Stage C does not affect the system. For example, if we double the capacity of Stage C, we can still produce only 100 widgets per hour. However, if we double the capacity of Stage B, we double the throughput of the system.

Critical Path and Critical Chain

For projects, the schedule constraint is the critical path. Experienced project managers recognize one major flaw in critical path: It assumes that you have no resource constraints. Experienced project managers have developed their own techniques for handling this situation. In 1997, Dr. Goldratt published another book based on TOC, called *Critical Chain.* In this book, Dr. Goldratt expands the concept of critical path by addressing this flaw and developing a more comprehensive approach to project scheduling.[2]

Let's consider a project. For simplicity, let's assume that we have three people on our project team. At some points in the project, three people are all that we can use and the project's schedule is constrained by the work itself. At other points in the project, we could use more people if we had them. But because we don't have more people, resources become the project's schedule constraint. This means that we could shorten the project schedule if we had more resources during that period. Critical path tells us how many resources we need; critical chain tells us when we'll really do the work because we won't get all those resources.

Dr. Goldratt's book covers many other project scheduling aspects, some of which I reserve for more advanced project management subjects. I present others here to expand the basic theory to the project portfolio.

Theory of Constraints Applied to the Portfolio

Portfolios exhibit similar characteristics to projects. After all, they are a collection of activities that require resources. The difference is that we can parallelize activities more in a portfolio than we can in a project. In projects, the mechanics of the work provide the structure of the workflow. In portfolios, other factors come into play.

Two constraints usually exist for portfolios. First the strategic plan dictates certain projects deadlines. For example, a goal in a camping supply company's strategic plan might require

that the organization increase its market share by 10 percent this year. To accomplish this goal, marketing determines that it needs to increase features in its stove product lines and add a new line for tents. The products must be ready for sale by April to capture the summer buying season. The strategic plan dictates project deadlines.

The second constraint is budget cycles. For example, the same company's strategic plan might state that the company wants to get four of its project managers PMP certified by the end of the year. No strategic time constraints exist; the only constraint is that budget money is allocated for this year.

Therefore, portfolio managers face two situations: projects that are time constrained and projects that are budget cycle constrained. The first situation defines the project from a time perspective. Understand that the end time of the project is fixed, not necessarily the start time. In our example, the products need to be in the stores by April. The project's start date is not relevant, as long as the products are in the stores in time for the buying season. This means we can start these projects early and spread them out to balance resources. Portfolio designers then can schedule budget-cycle projects during otherwise quiet times.

Finally, we want to maximize the use of our internal resources, but we recognize that we can outsource when needed.

When we apply the infinite resource pool concept, resources are no longer a constraint—they become a design choice.

This gives portfolio designers significant flexibility in scheduling projects. The portfolio designer's goal is to level the workload at the organizational level across the portfolio, ensuring that the organization completes time-constrained projects on time. To accomplish this goal, we use throttlers.

Synchronizers and Throttlers

A company hired me to help them implement project management. Although the organization describes itself as a publisher, it distributes most of its products in electronic format, which requires custom software. The company claims to have 17 products, and it upgrades each product every year. It releases all new products and product upgrades at the end of November.

> My experience in the software industry enabled me to determine the company's primary constraint quickly. I asked, "Where is your software testing department?"
>
> The answer: "She sits over there."
>
> The company had many software architects and programmers, but only one tester. This poor lady had nothing to do for 10 months out of the year.

Then for 2 months, she had to perform testing on 17 different products. She was the throttler for the entire company.

A throttler is the individual, group, or department that limits the throughput of the projects in a portfolio.

In this case, we didn't need to outsource to complete projects on time; we simply had to stagger the projects throughout the year based on her ability to test the software. Problem solved.

Identifying Throttlers

When we apply the theory of constraints to a portfolio, we postulate that, at any one time, one department is the resource constraint. In the previous example, it is software testing. In general, portfolios are complex systems, involving many different kinds of projects spread across the different organizational units. Despite this, locating the throttler might not be as difficult as you imagine.

Any experienced manufacturing manager knows how to find the throttler in a manufacturing plant: look for the largest pile of work in progress. In our earlier widget example, Stage A can do its job much faster than Stage B. Stage A's output stacks up in front of Stage B. Stage C can also do its job faster than Stage B, so as Stage B finishes its job, the work can flow immediately through Stage C. Little or no work in progress is sitting in front of Stage C. All the work in progress is stacked up in front of Stage B—it is the throttler or the constraint.

Locating the portfolio throttler in an organization is similar. With the electronic age, we can no longer look for the group of people with the deepest stack of paper in their in basket, but the project managers know where the constraints are. They're the departments that are always overworked, that are always working on the weekends, and that the project managers complain about the most. They are also the ones crying out for help—if their managers would only listen.

I was conducting a seminar at a military depot. I asked the class for their single biggest problem of getting projects done on time. Their answer was "Building 17." Building 17 is where they conduct final inspections. The question was simple and the answer was both immediate and unanimous (note that I changed the number of the building and function it performed, to protect the guilty). Work flowed nicely until it reached Building 17, where it waited for inspection. Equipment was stacked up outside the building, sometimes for weeks.

If you want to identify your throttlers, ask your project managers.

Synchronizing Projects

When we know the location of the bottlenecks, portfolio designers can design an efficient and successful portfolio. First, we must do the following:

- Synchronize projects to level the resources within the constrained departments
- Target those functions for potential outsourcing
- Conduct cross-training to aid workflow into and out of those departments

Nonbottleneck Departments

If one department is overworked and throttling projects, does that mean that the other departments are underutilized? The answer is "Probably." Before the economic cataclysm that occurred in 2008, I made this bold claim in most of my seminars:

Most organizations are overstaffed.

With mature project and portfolio management, you can accomplish the same amount of work with fewer resources. This is, after all, the primary theme of this book.

You face the problem that, despite the strategic planning committee's and portfolio designer's best efforts, different departments are busy at different times. Accounting departments get busy at the end of each quarter and busier at the end of the fiscal year. Marketing departments get busy during product design and rollout, but are less busy in between. Create a solid strategic plan, train talented portfolio and project designers, and watch your organization's output soar.

Deliverables-Oriented Assignments

In the beginning of this book, I indicated that work produces a deliverable, and the deliverable achieves the goal. We need the deliverable to achieve the goal, not the work. Moving people off one task and onto another when they haven't finished the first task is highly inefficient. The deliverable is not done—the work cannot achieve its goal without the deliverable.

Ronny's Resource Dilemma

Ronny is a project manager for software development effort. His company just purchased a software package that helps design and manage business processes. His project is to integrate this package into the organization's accounting system to simplify financial reporting and to help the company maintain compliance with the Sarbanes-Oxley Act.

Ronny has two systems designers working on various system components. Ronny has assigned the work package that creates the accounting interface module design to Karen, one of the organization's best designers. The duration estimate for the work package is two weeks. When Karen is finished, she's scheduled to move to another project within IT, run by Carol. That work package is on Carol's critical path; therefore, it's time critical.

When Karen starts work on Ronny's work package, she discovers that the organization made changes to the accounting system since she estimated the task. She warns Ronny that the work package might take longer than two weeks. Ronny warns Carol and tells Karen to do her best to complete the work package on time. Karen keeps Ronny apprised of her progress every few days.

Several days before the scheduled end date, Karen formally informs Ronny that she cannot meet the deadline.

For this example, we need to assume that we have immature project management because we made many mistakes so far. Let's assume that Karen is the only person who can perform both work packages (shame on us for not cross-training and not having a designated backup), that we can't outsource the work (shame on us for not developing a strategic outsourcing plan), that both work packages are on the critical path (shame on us for scheduling a single resource on two critical path activities on two different projects), and that we have not developed any schedule contingency (shame on us for not managing risk or throttling our projects).

That leaves us with the question "At the end of the two weeks, does Karen move to Carol's project, or stay and finish Ronny's?"

The answer is a clear and resounding "She stays."

Moving her off the first project effectively negates the work she did. If Ronny brought in someone else, that person would need to get up to speed on the project and Karen's work, and then finish the work. If Ronny simply waited for Karen to finish Carol's assignment and then come back, she would also need to get reacquainted with the project and the work she had done so far. The time lost on Carol's project would be less than the time lost on Ronny's.

Resources should move between assignments based on when they finish their work, not on a predetermined schedule.

Make sure that your people finish their jobs and create their deliverables before you pull them off to do something else.

Establishing a Resource Management Doctrine

We now have the tools to make a quantum leap in organizational efficiency. By implementing a few simple tools and developing a sound doctrine, you can take your organization to the next level. While these tools are simple, you must be thoroughly understood.

A glib approach to achieving organizational efficiency will fail.

Organizations are complex systems. Organizational agility, portfolio agility, project agility, and a highly efficient organization are not developed overnight or through superficial

understanding. An automobile is a complex system, yet you find it easy to control and manage an automobile because they have been engineered over a very long period of time and all its parts work closely together to achieve your primary goal: to transport you, your passengers, and your cargo safely from one location to another.

However, that's not the complete picture. Your car's designers included features to handle an ever-changing environment. You can drive through clear, rainy, or snowy weather without stopping. You can drive through cold or hot weather. The designers included features to help you avoid accidents. They specifically designed the center of gravity, the suspension system, and the tires to prevent the car from rolling over during sudden maneuvers. They included airbags and seat belts for risk mitigation, in case you can't avoid the accident.

Globalization, the Internet, and open markets have created an ever-changing environment. Your organization's capability to survive depends on all its pieces working together and adjusting to the dynamic environment.

Let's recap our strategy.

Utilization

Remember, if your staff is 100 percent utilized, 100 percent of your projects will be late. Different organizations have different optimum utilizations. Each department within an organization also exhibits unique optimum utilization percentages. An engineering department can be very efficient if the engineers are left alone to design instead of being distracted by handling emergencies. Researchers require more creative time to develop different ideas.

Determine the appropriate utilization percentage for your organization, and have each department determine their own. Overworking your staff doesn't improve their efficiency or your organization's production. Better yet, drop utilization as a key metric altogether. Instead, devise a metric that measures value-added or products per unit time instead of work per unit time.

Identify Background Activities

All organizations have activities that need to be accomplished but aren't time critical. All organizations also have resources that need to jump onto a time-critical activity as soon as the activity is ready or to have some productive distractions. Managers can assign background activities to staff who might be between assignments or who need to be ready to begin a time-critical activity when it's ready. This keeps the resource available, keeps them productive, and gets the background activities done, all at the same time.

Balance Single-Tasking and Multitasking

Some disciplines require concentrated effort; others do not. Some people are good at multitasking; others are not. Strike the right balance between single-tasking and multitasking.

Cross-Train

Cross-training not only improves efficiency, but it also enables you to move people from one function to another when resource needs shift.

Define Time-Critical Activities

Project managers find the time-critical activities using critical path methodology. Functional and process managers must also determine time-critical activities. When you have identified these activities, you can schedule resources more efficiently, ensuring that you can source your time-critical activities quickly.

Schedule Resources Based on Deliverables, Not a Calendar

The objective of work is to create a deliverable that achieves a goal. The deliverable is important. Pulling a resource off a task before it's complete is inefficient and might prevent the resource from completing that deliverable. Schedule resources so that they can finish their deliverables. Then you will realize your goals.

Summary

Resource efficiency is not difficult, but it does require an intelligent, systematic approach. Working your staff hard doesn't necessarily achieve your goals; working them right does. Table 15.1 defines the organizational evaluation matrix for this chapter.

Table 15.1 Organizational Evaluation Matrix

Performance Item	Organization's Performance High ← → Low				
My organization uses a systems approach to developing its resource management doctrine.					
My organization has an established baseline for optimum resource utilization, and each department has its own baseline for optimum resource utilization.					
My organization ensures that resources needed to perform schedule-critical activities can start as soon as the activity is ready.					
All project managers establish a critical path or critical chain for their projects.					
My organization balances single-tasking and multitasking to take advantage of each.					

Performance Item	Organization's Performance High ← → Low				
My organization cross-trains its resources to improve workflow and enable resources to shift roles with resource demands.					
My organization has a method to determine resource constraints at the portfolio level.					
My organization schedules projects to balance resources at the portfolio level.					
My organization's management encourages staff to finish their work before moving them to other work.					
My organization bases efficiency on organizational goals achieved, not just work performed.					