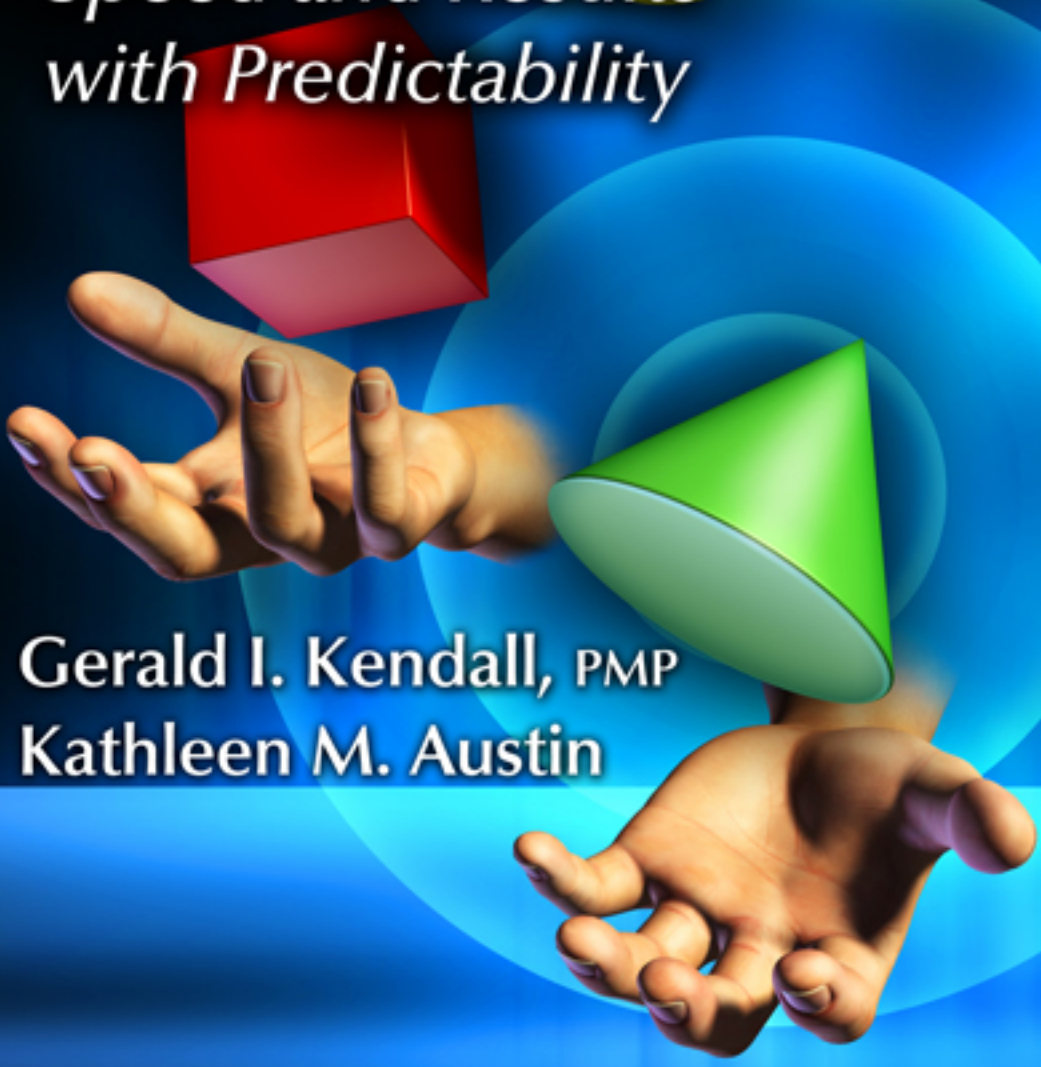
ADVANCED

Multi-Project Management

Achieving Outstanding Speed and Results with Predictability



Gerald I. Kendall, PMP
Kathleen M. Austin

23. Three Points of Network Insulation

The Premise

When we talk about insulation in project terms, we mean strategically protecting the work of the project network in order to enable delivery of the project on or before the due date, at or below budget, and with full project scope.

There are three points in any project network that are logical places to insulate a project against variability. The first is the project due date. The second is any point that feeds into the project's critical chain or critical path. The third (which does not exist in all projects) is any point that represents a critical milestone date to the project.

Over the past 20 years, we have found the most effective method of insulating projects is with a few buffers of time. This chapter shows the details behind the buffer calculations and placement. The software referenced in Chapter 10 helps to model buffers and reduce the amount of manual effort required. You do need to understand the underlying reasoning behind the buffer calculations.

The Example

As we saw in Chapter 21, the critical tasks within a project represent the longest pathway of task and resource dependencies in the project network.

In the project network shown in Figure 23.1, each box is a task, task names are the letters inside the box, and the resources are represented by the patterns in the boxes. The ambitious and standard time estimates are shown below each task. There is one of each resource type available. The critical tasks, tasks H, J, B, C, D, E, F, G, O, P, and Q, are shown with a bold outline. This example uses the critical chain approach for identifying the most critical tasks within the project. For more information on this approach, see the books listed in the Bibliography. If you are not familiar with this approach, you may use a similar approach with the critical path methodology. If you are not familiar with either methodology, you may simply assume that these tasks were identified as the ones most likely to determine the duration of the project.

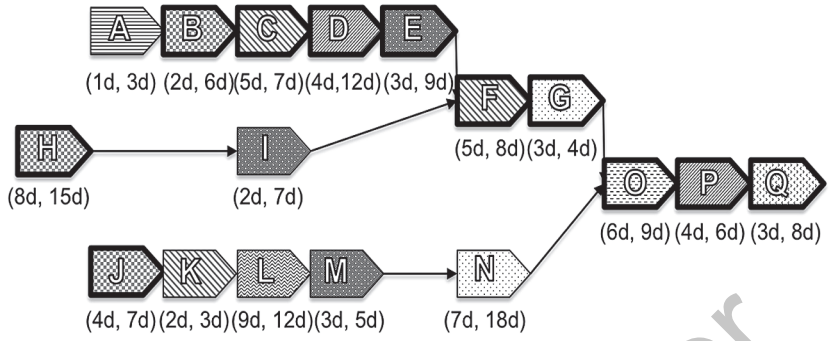


Figure 23.1 Project network with task names (letters), resources (patterns), time estimates (ambitious, standard), and critical tasks identified (bold task outline: H, J, B, C, D, E, F, G, O, P, Q)

The critical tasks are determined using the ambitious task times for each of the tasks in the project; this does not consider the inherent variability of each task. Ignoring variability on paper does not make it non-existent in a project, a fact unfortunately learned and relearned by many project managers. What we need is a way to understand and accommodate variability while planning and scheduling a project; during execution we need a way to understand when and where it occurs, as well as its impact on project completion. The variability we're talking about is the difference between the standard and ambitious task times. For future reference, we'll call this difference in task times the *variability factor*.

The problem with using only standard time estimates for planning, scheduling, and project execution is that each task attempts to account for variability individually. This is comparable to an insurance company trying to protect only one house, with a small premium. If the house burns down, the single premium will not have a hope to cover the losses. It only makes sense to protect against losses through aggregation—insuring a lot of houses where the loss is likely to occur only in a small percentage.

This same principle is used with these task time estimates. The variability factor is removed from the task time, leaving each task with an ambitious time estimate (a small premium). The variability factor is then cut in half, and the remaining half is aggregated into time buffers that insulate the network (protecting against *losses* or variability that will occur in some, not all, tasks).

Some organizations advocate for more sophisticated approaches to determining the size of project buffers. In our experience, 90% of all organizations will gain sufficient benefit from the inherently simple approach described above. Furthermore, once the organization has buffers and is monitoring these buffers

during execution, it often finds that there are other much more important leverage points to further improve multi-project management.

We primarily use two types of time buffers to provide the needed network insulation:

1. The project buffer protects the project's due date from variability within and along the critical tasks. There is one project buffer per project.
2. Feeding buffers protect critical tasks from being delayed by variability along and within the non-critical feeding pathways.

Why and how are these two insulation points different and important? Traditionally, variability is embedded in each task and is not visible or managed in a project (typically one task time estimate is used—the equivalent to standard).

In addition, having the variability embedded motivates the wrong behavior in both leadership and resources during execution. Many times during project execution it seems that the task estimates somehow become deterministic numbers, translated into *due dates* for each task. Task due date compliance then appears to be a valid measure of not only project progress, but also employee *goodness*. What happens if the resource performing the task does not run into a lot of problems and actually is able to achieve the task completion criteria at the more ambitious time or even sooner? Is there any motivation to announce completion early? On the contrary! There are more reasons to NOT announce early completion:

- Many people doing project tasks fear that similar task estimates will be cut in the future, increasing their personal risk of not meeting the task due date compliance measurement.
- Others see holding on to the task as a way to have time (or budget) to accomplish other necessary project or non-project work.
- Sometimes there is a view that the work quality is not good if the resource takes less time than allocated. Also, knowing they are being evaluated according to task due date compliance, resources willingly allow work to *expand* to fill the time available (known as *Parkinson's law*).

The result of these motivations: almost no tasks are turned in by resources earlier than the standard task estimate and many finish later. Since actual project lead time is made up of the critical task time completions, it doesn't take many late finishes to equal a late project.

There is a better way to plan and schedule a project that enables successful project execution and management! The *best process* we've found for understanding and accommodating task variability is to gather two task time estimates

(as was shown in Chapter 20) and use the understanding gained to protect or insulate the project strategically in the schedule and for execution. Then during execution, monitor and manage the use of the insulation. These concepts will be discussed further in upcoming chapters.

Project Buffer

A project has only one project buffer (shown in diagrams below as PB). Sometimes, a project has critical milestones. In this case, a portion of the buffer can be used for monitoring these milestones. Further explanation is provided below. The project buffer is placed between the last critical task and the project's due date (see Figure 23.2).

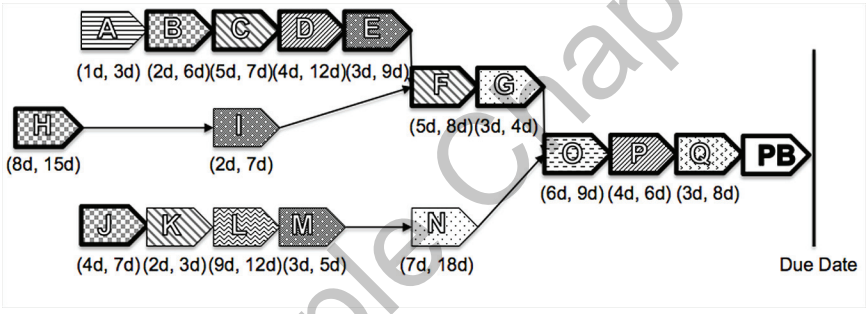


Figure 23.2 Placement of project buffer (PB)

The two important criteria for any project buffer are ensuring proper placement and proper sizing:

- *Placement.* The right side of any buffer is always placed directly before whatever it is protecting: for the project buffer, that is the project's due date. The placement is viewed as strategic for the project because the project buffer acts both as a *shock absorber* and a *time bank account* protecting the due date from variability in and along the critical tasks.
- *Sizing.* Ensuring the project buffer is properly sized allows it to act effectively as a shock absorber and time bank account. However, the project buffer is a unique type of shock absorber, since it can grow in size when critical tasks finish earlier than planned. In its traditional role, negative task variabilities are cushioned and protect the due date; a time bank account in that deposits (positive task variabilities) and withdrawals (negative task variabilities) can be made without negatively impacting the due

date—assuming the project buffer is being properly managed during project execution.

- At the end of the project, if the bank account has a positive balance, it means that the project finished earlier than planned. Sometimes, the organization can take advantage of this by releasing the final project output to the customer earlier than the promised due date. Sometimes this is not possible. However, even in the latter case, there is still a major positive impact of finishing earlier—all of the resources are freed up sooner to work on the next project.
  - Proper sizing of the project buffer starts with evaluating the variability of the critical tasks arithmetically. The typical rule-of-thumb arithmetic starting point for the project buffer is identifying the variability factor of each task time (standard minus ambitious), summing it up for the critical tasks, and then dividing by 2. In some environments, we recommend then adjusting (almost always upward) for specific, documented risks. Since readers of this text come from many different environments, we do not know whether there is a need to adjust the buffer and if so by how much.
  - In Table 23.1, you will see the calculation for the project buffer as 22 days, using the simple formula described above. Having now used this variability information for the critical tasks, you will see in Figure 23.3 that we are now showing only the ambitious times,

Table 23.1 Project buffer calculation

Task	Ambitious	Standard	Variable Factor VF = S – A
H	8	15	7
J	4	7	3
B	2	6	4
C	5	7	2
D	4	12	8
E	3	9	6
F	5	8	3
G	3	4	1
O	6	9	3
P	4	6	2
Q	3	8	5
			Total = 44
			½ VF = 22

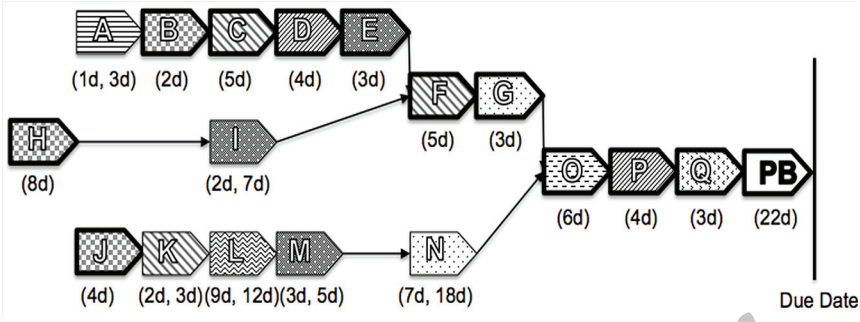


Figure 23.3 Properly sized and placed project buffer (only ambitious times are shown for critical tasks)

which are the times we will monitor as we execute the project. Since we have not yet calculated buffers for non-critical tasks, those tasks still show both the ambitious and standard times.

Feeding Buffers

Wherever tasks feed into the critical tasks, there is a danger that the feeding path will be completed late, causing an interruption in the critical tasks. To insulate against this variability, a time buffer called a *feeding buffer* is used. Each feeding buffer is placed with the origin (right side) up against whatever critical point it is feeding.

In Figure 23.4, there is:

- A feeding buffer between non-critical task A and critical task B
- A feeding buffer between non-critical task I and critical task F
- A feeding buffer between non-critical task N and critical task O

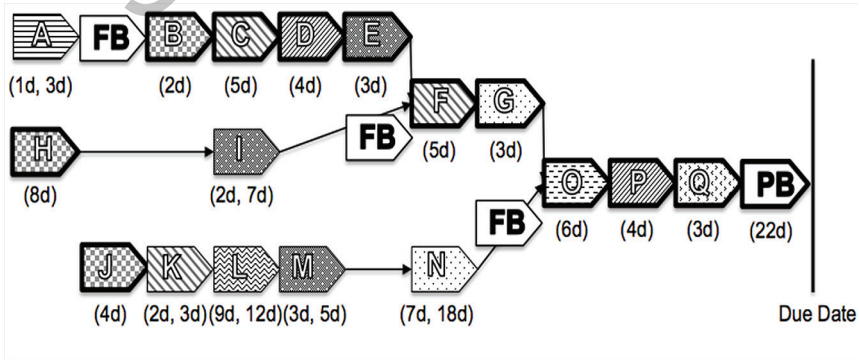


Figure 23.4 Properly placed feeding buffers

Table 23.2 Feeding buffer calculations

Feeding Buffer Projects	Feeding Chain Tasks	Ambitious	Standard	Variable Factor (VF) VF = S – A	Feeding Buffer Size (VF/2)
B	A	1	3	2	1
F	I	2	7	5	2.5
O	K	2	3	1	8.5
	L	9	12	3	
	M	3	5	2	
	N	7	18	11	

Sizing feeding buffers follows the same formula as the project buffer (see Table 23.2). Think of the feeding buffer as being the first line of defense against variability for non-critical tasks. Their second line of defense, although not preferred, is the project buffer. Figure 23.5 shows the entire project plan with ambitious times.

Iteration Variability and Buffer Sizing

As discussed in Chapter 20, iteration variability can occur along both the critical chain/critical path and along non-critical paths. When sizing either the project buffer or feeding buffers, iteration variability (where it exists) must be included in the original calculations. The ambitious number of iterations is already included in the pathway's buffer calculations (since the ambitious number of iterations is shown in the pathway). See Figure 23.6.

What remains is accommodating the variability of the potential additional iteration(s). Assuming the task times remain the same in all iterations, the buffer

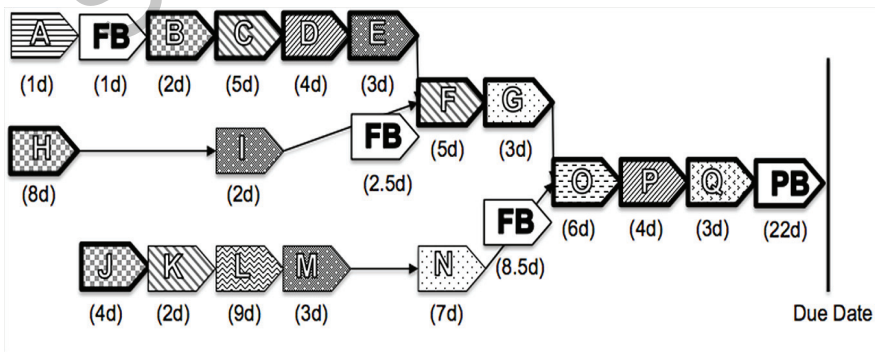


Figure 23.5 Properly sized and placed feeding buffers (only ambitious times are shown for all tasks)

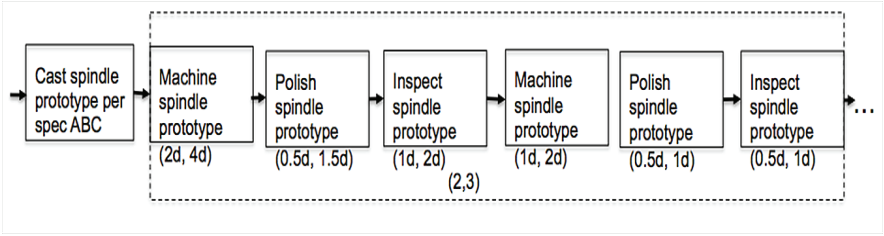


Figure 23.6 Iteration variability: ambitious 2 iterations, standard 3 iterations (2, 3)

adjustment is calculated by taking one half of the sum of the variability factor times the number of additional iterations. For example, if all the tasks within an iteration accounted for 10 days ambitious, and 22 days standard, then each additional iteration has 12 days of variability. Half of that, or six days, would be added to the appropriate buffer for each possible additional iteration.

Additional Network Insulation Point

There are a few cases where an additional network insulation point is needed within a project; it is referred to as a *contractual* or *critical milestone* (CMS); there are two types that can exist. Please note that CMS use should be avoided as much as possible; it often increases a project's lead time.

- *CMS type 1.* Portions of project work must be accomplished by a specific date, where a review (sometimes a *stage gate* or *milestone review*) must occur. All work stops until the review has been completed and approval is given (often via funding) to proceed with the project. The CMS date is protected by a time buffer called a *CMS buffer*. The CMS can occur along a critical path or a non-critical path.
  - *Modeling the CMS.* The task is a milestone (duration 0, 0) in the network but has a start date attribute associated with it—the date the review starts.
  - *Placing the CMS buffer.* The CMS buffer (time buffer) is placed between the last task dependency and the CMS.
  - *Sizing the CMS buffer.* The tasks that must be included for sizing this buffer are the longest string of task dependencies leading to the milestone. These task dependencies can include both critical and non-critical tasks.

  - *Impact on other buffers.* This type of CMS buffer actually *un-aggregates* the safety for the feeding or project buffer with which it is associated. When this type of CMS buffer is used, the associated feeding or project buffer must be adjusted.
- *CMS type 2.* Some projects require interim deliverables tied to a specific date. For example, a prototype must be delivered on or before a certain date. That date must be protected by a CMS buffer, but this type of CMS does not involve stopping project work to wait for approval/go-ahead; for this type of CMS, the tasks after the CMS can begin as soon as the tasks before the CMS are completed.
  - *Modeling the CMS.* The task is a milestone (duration 0, 0) in the network but has a start date attribute associated with it—the date the review starts. However, this CMS is not placed in-line. Typically, it is shown above the line of tasks, as its associated date is *external* to the remaining work of the project.
  - *Placing the CMS buffer*. The CMS buffer (time buffer) is placed between the last task dependency and the CMS, in this case not pushing the next project task.
  - *Sizing the CMS buffer*. The tasks that must be included for sizing this buffer are the longest string of task dependencies leading to the milestone. These task dependencies can include both critical and noncritical tasks. Remember: we are sizing this buffer to protect the CMS date.
  - *Impact on other buffers*. This type of CMS buffer does not *un-aggregate* the safety for the feeding or project buffer with which it is associated. The associated buffer does not need to be recalculated.

Conclusions

When projects are strategically protected with buffers of time at three points, they become much better able to achieve their outcomes. The problem in most environments is that protection is left at an individual task level, which almost guarantees failure. By removing the protection against variability that is inherent in most task time estimates, and moving it to aggregated buffers, the protection is in three places where it makes sense and can be carefully monitored as the project executes. This chapter provides the details behind how to calculate the three types of buffers and where to place them.

Questions

23-1. What are the primary two points in a project network that must be insulated?

23-2. What is the primary determinant of how long a project will take?

23-3. A variability factor is aggregated into what type of buffer?

23-4. What does the project buffer protect?

23-5. How many project buffers can a project have? Why?

23-6. A properly sized project buffer allows it to effectively do what?

23-7. What do you do with iteration variability when sizing buffers?

23-8. Why should you try to avoid CMS buffers?

24. Operations versus Project Responsibility—Resource Insulation

The Premise

The last chapter talked about insulating projects from variability. You may be asking yourself, "What is 'resource insulation' and why is it important?" Resources, whether people, facilities, or capital equipment, are available in limited quantities. In many organizations, the same resources that perform project tasks also have non-project work responsibilities. This non-project work is crucial for the organization's operational (day-to-day) success. This is another, often major conflict for project and resource managers as well as operational managers! This chapter uncovers some successful approaches for us to protect or insulate our resources so that all the required work can be accomplished.

Negative Effects of Multitasking

A common practice in many organizations is to interrupt resources assigned to a project task to go work either another project task or perform operational work, as priorities change. What happens, in terms of task duration, when task work is interrupted before it is completed? The clock keeps ticking on the task duration, but no work is being performed—the task is *suspended*. When task durations are extended because of *suspend* time, project durations are as well.

Task times are unlikely to be adequate when resources are interrupted frequently from doing project work or are often busy doing other work. If the task times are not adequate, project scope, budget, and due date are quickly threatened.

When a resource working on a project task is interrupted, it often requires substantial time to pick up where they left off. Time is needed to get set up again and/or review what is needed and what has been done already to know where to start. The result is that the task times are extended again, not because of task variability (which is how the task times were estimated), but because of the

multitasking. According to *Harvard Business Review*[1] and the *New York Times*,[2] "when you switch away from a primary task to do something else, you are increasing the time it takes to finish that task by an average of 25 percent."

Our experience shows that the amount of damage from multitasking varies, depending on the nature of the task. Tasks that require a great deal of concentration, such as some engineering or IT architecture or debugging tasks, suffer much more than the 25% time damage. They also suffer from mistakes and rework.

You will have major gains in task durations and project predictability by fully stopping the multitasking of resources on project tasks. We are not saying that resources should be assigned to a project and not asked to do anything else for the duration of the project. We are recommending that when a resource has started a project task, there be no switching work until that project task is finished (another good reason why clear and precise task definition is crucial!).

Buy-in to this concept (no multitasking on project tasks) can be difficult, sometimes more difficult for the resources than for management. There are some resources that seem to like the security of having several tasks open for them to do. Some will say they need variety for when they are stuck. We have found that training and clear task definitions as well as management commitment will permanently resolve this issue. The positive results to project and non-project work are immediate. Multitasking is by far one of the biggest wastes of limited, available capacity in most multi-project environments today!

Planned versus Actual Resource Loading

During planning and scheduling, remember that you have used ambitious times for the tasks modeled with resources identified. The rest of the time allocation (a third of the total time) went into buffers and was not directly associated with a specific resource skill set. Since most tasks will not complete in the ambitious time or less, it requires a different perspective when evaluating resource loading. We expect some tasks to take longer than the ambitious time and some to take less; some tasks may even exceed their standard time. The planned time use of the resources with only ambitious times means that *fully loaded* must be viewed as much less than 100%.

Resource histograms (see Figure 24.1), when using ambitious times, will also be very sensitive to workday calendars. If resources are expected to be *at work* for eight hours per day, does that really mean they are expected to perform project work for the entire eight hours? What portion of their day is tied up with meetings, phone calls, e-mails, and administrative tasks? In some organizations, the available project work time can be four hours or less per day per resource.

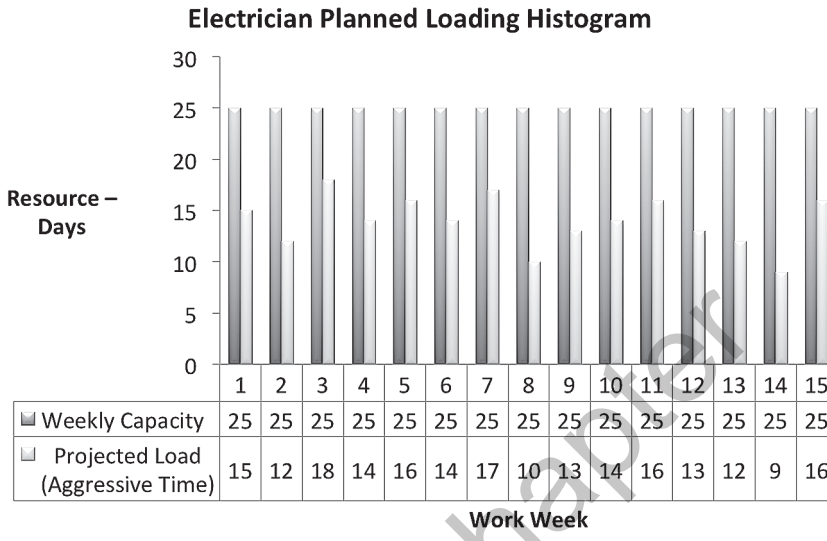


Figure 24.1 Resource loading histogram using ambitious task times

Consider carefully and document fully so that anyone using a resource loading histogram understands what they are looking at before making any resourcing decisions.

Let's say a resource is planned to be available for project task work four hours a day. If the task they are assigned to work is estimated to be (2d, 5d), does that mean they will spend at least two four-hour days working the task or was the task estimated with an eight-hour day? If the task was estimated based on an eight-hour day and the resource does four hours of project work per day, expect the resource to be tied up for at least four days (16 hours). Is the resource fully loaded at four hours a day for project work? Calendars and histograms need to be fully understood to have proper resource insulation!

Supporting Projects and Operations with the Same Resources

Depending on the organization's environment, these techniques work effectively, along with no multitasking, when the same resources are needed to support both projects and operations:

- Determine a way to prioritize the operations work and the project task work in the same system. At the beginning of each workday, assign

available resources in priority work order. At the end of each day, ensure task and operations updates are done so that the priority list is ready for the next day, as well as the list of available resources. If either operations work or a project task completes before the end of the day, assign the available resource(s) to the next highest priority work.
  - *Positive(s):* All work is prioritized so that operations as well as project tasks are accomplished.
  - *Negative(s):* It can be difficult to merge priority systems.
- Allocate the resources so that some people primarily work operations and the remainder primarily work project tasks. Typically, this is done on a rotation basis. For example, the resources work three months of dedicated availability for operational work, then are available for three months of dedicated project tasks.
  - *Positive(s):* The resources can focus fully on building operational skills; when required for overload situations on projects, these resources can be made available on a dedicated basis for a short time.
  - *Negative(s):* There are times when there is only one of a resource skill, so the resource must work both; there are no other options.
- Dedicate a portion of the workday to operations work and another portion of the workday to a project task.
  - *Positive(s):* Both are being accomplished.
  - *Negative(s):* Both will take longer since the daily work calendar for both operations and project work has been shortened.

Guaranteeing Subcontractor Availability

We have seen organizations use a contractual agreement (with financial compensation) for an external resource to guarantee its availability when needed to perform critical tasks. The project manager must give updates to the resource as the time nears for that work to begin. As an example, on a road construction project, a two-lane paving machine is required for a specific critical task. It does take time to move the paver from one location to another, and the project manager does not want to delay critical work waiting for it to arrive and be set up. In this case, part of the contract for the resource included a negotiated additional fee guaranteeing the paver would be available to work based on the previous task's completion updates. Note that the paver did not agree to be available on a certain date in advance; rather, there was a window of time that it was likely to be needed and then, based on task updates while the predecessor task was executing, a countdown was given to the paver. Warnings:

- Use contractually guaranteed resourcing only when the benefits outweigh the additional costs.
- Do not enter into a contractual agreement with resources that are not capable of fulfilling their part of the deal!

Conclusions

The conventional way of dealing with people who have both operational and project responsibility is to accept the conflict between the two as a fact of life. This chapter describes the proven damage caused by such acceptance and shows different ways to permanently mitigate it. One way is to dedicate an operational resource to a project for the entire duration of a single project task, delegating their operational responsibilities for this duration to someone else. Another approach is to allocate a portion of a day to operations and a portion to projects, thus at least minimizing the amount of damage caused by interruptions. A third way is to allocate a portion of a resource pool to operations while dedicating the remainder to projects. No matter which way is chosen, it is vital to get out of the mindset that it is OK to continually multitask.

Endnotes

1. Tony Schwartz, "The Magic of Doing One Thing at a Time," *Harvard Business Review*, March 14, 2012. Retrieved from http://blogs.hbr.org/schwartz/2012/03/the-magic-of-doing-one-thing-a.html.
2. Steve Lohr, "Slow Down, Brave Multitasker, and Don't Read This in Traffic," *New York Times*, March 23, 2007. Retrieved from http://www.nytimes.com/2007/03/25/business/25multi.html?_r=1&pagewanted/%20all=&pagewanted=print.

Questions

24-1. What is resource insulation?

24-2. What can happen to project duration when tasks are *suspended*?

24-3. Are project and feeding buffers sized to accommodate resource non-availability?

24-4. When a resource comes back to a project task, can it just pick up where it left off?

24-5. Multitasking wastes what?

24-6. Buy-in to the concept of monotasking (instead of multitasking) on project tasks can be more difficult for whom?

24-7. Why should contractually guaranteed resourcing be used sparingly?