9

Project Scheduling

Networks, Duration Estimation, and Critical Path

Chapter Outline

Chapter Objectives

After completing this chapter, you should be able to:

1. Understand and apply key scheduling terminology.
2. Apply the logic used to create activity networks, including predecessor and successor tasks.
3. Develop an activity network using Activity-on-Node (AON) technique.
4. Perform activity duration estimation based on the use of probabilistic estimating techniques.
5. Construct the critical path for a project schedule network using forward and backward passes.
6. Identify activity float and the manner in which it is determined.
7. Calculate the probability of a project finishing on time under PERT estimates.
8. Understand the steps that can be employed to reduce the critical path.

PROJECT MANAGEMENT BODY OF KNOWLEDGE CORE CONCEPTS COVERED IN THIS CHAPTER

1. Plan Schedule Management (PMBoK sec. 6.1)
2. Define Activities (PMBoK sec. 6.2)
3. Sequence Activities (PMBoK sec. 6.3)
4. Estimate Activity Resources (PMBoK sec. 6.4)
5. Estimate Activity Durations (PMBoK sec. 6.5)
6. Develop Schedule (PMBoK sec. 6.6)
7. Control Schedule (PMBoK sec. 6.7)

PROJECT PROFILE

After 20 Years and More Than $50 Billion, Oil is No Closer to the Surface: The Caspian Kashagan Project

Two decades ago, the world was in desperate need of new sources of oil, just as emerging economies were anxious to exploit their natural resources in exchange for improvements in the standard of living. It was against this backdrop that the Kashagan oil project was launched. A partnership between Kazakhstan and a consortium of oil exploration companies (including Shell, Exxon Mobil, Total, ConocoPhillips, and Eni, among others), the Kashagan project involved offshore drilling in the Caspian Sea. The oil field was discovered in 2000, and with oil reserve estimates that are said to be the largest in the world outside of the Middle East, the plan was for oil to begin flowing in 2005, with a projected daily output of 1.5 million barrels. One Shell driller labeled the field "an Elephant."

Now, years behind schedule and with a budget that has grown from its original total estimate of $57 billion to $187 billion, the project is still far from completed. Phase 1 of the project was expected to cost $24 billion and the bill has already grown to $46 billion with little to show for it. In addition to its massive budget overruns, the project has been continuously plagued by a series of engineering missteps, management disputes, miles of leaky and corroded pipelines, and technical problems. So bad has the situation become that the project has now been halted indefinitely while all parties try to understand what went wrong and how to get things back on track.

The Kashagan project's problems come at a time when relationships between Western oil companies and resource-owning governments are more important than ever. To replace what they pump, oil companies need to collaborate with state-owned companies that control 90% of the globe's remaining oil reserves, by a World Bank estimate.

FIGURE 9.1 Kashagan Oil Field

Source: Shamil Zhumatov/Reuters/Corbis

But governments often give foreign oil companies access only to the hardest-to-develop acreage. Kashagan's disastrous overruns show how these "public/private" collaborations in difficult oil fields can quickly go bad for both sides.

Kashagan is a complicated project under the best of circumstances. The oil derricks sited offshore had to be redesigned atop "islands" that were built for them when it was discovered that the Caspian is shallow enough at this location that it freezes all the way to the bottom during the winter, making drilling with conventional rigs impossible. The companies had to build artificial islands of rock and rubble and drill through these. Further, the oil is under high pressure from corrosive natural gas high in toxic chemicals. That meant building a sulfur-removal system onshore, reached by a pipeline, for the portion of the gas to be recovered. It took operators nearly two years to factor this level of sour gas into infrastructure design. And heavy pipe-laying machines sometimes broke down in the cold.

A number of circumstances have contributed to the schedule delays and budget overruns, including:

- Administrative confusion as the major contractors could not decide who would be in charge of the development. When Exxon Mobil originally attempted to take charge, Shell executives threatened to pull out of the partnership. Ultimately, the much smaller Eni SpA was named lead contractor, although each company had veto power over all major planning decisions.
- Relationships with the Kazakhstan government have deteriorated as the project has experienced delay after delay. By 2008, the government began levying penalties for the extended delays, making the oil companies' investment in the project all the more expensive. Eni Chief Executive Paolo Scaroni said his company's relationship with the government "has been excellent" considering the years of trouble. A senior official of Kazakhstan's state-owned oil company, KMG, disagreed. "It's a marriage that is made in hell," he said.
- Problems with human resources assigned to the project. As part of the agreement with the local government, oil companies had to employ large numbers of local workers, with a portion of them mandated to perform office functions. One former official recalled hiring hundreds of enthusiastic locals who had "never sat in front of a computer."
- Leaking pipes. In 2013, the companies prepared for a milestone: starting commercial oil production and transferring the role of operator to a Shell-led group. On September 11, the companies announced that oil was flowing. About two weeks after that, parts of the underground gas pipeline began leaking into the Caspian Sea. Oil pumping stopped while workers inspected and found a leak. Crews patched it, and oil pumping resumed. Two weeks later, the pipeline sprang new leaks. This time, the companies shut down the whole Kashagan operation. Workers spent the fall excavating parts of the 55-mile pipeline and sending section for tests at a UK lab.
- Technical errors. The oil companies used outdated Russian cruise line ships as floating barracks for oil workers. However, besides these construction-worker barracks, offshore accommodations for the permanent staff were needed. Around 2005, Eni's partners realized that plans for these put them too near a production site. Eni redesigned the accommodations, delaying construction by another year.

The series of missteps and technical challenges has left everyone feeling dissatisfied. Oil company and Kazakh officials sniped at one another. "Nobody's happy with the governance, and I don't think anybody's happy with the operatorship," Shell Chief Financial Officer Simon Henry said. Cracks were found in several places along the pipeline, according to people familiar with the inspection, who said it appeared the metal had lost some of its factory characteristics, possibly through a combination of poor welding practices and the natural gas's hydrogen-sulfide content.

Finally, the combination of technical challenges, unexpected (and unresolved) pipeline leaks, finger-pointing by executives from Kazakhstan and the oil companies, and stiff financial penalties for delays became too much for some of the consortium. In 2013, longtime partner ConocoPhillips sold its stake in the project to KMG, which later resold it to China National Petroleum Corp. "We got our $5.5 billion in the bank and got out of Kashagan," said Al Hirshberg, a Conoco executive, at a conference last fall. He added: "It feels good to be out of it."[1]

INTRODUCTION

Project scheduling is a complex undertaking that involves a number of related steps. When we think about scheduling, it helps if we picture a giant jigsaw puzzle. At first, we lay out the border and start creating a mental picture in our heads of how the pieces are designed to fit together. As the border starts to take shape, we can add more and more pieces, gradually giving the puzzle shape and image. Each step in building the puzzle depends on having done the previous work correctly. In a similar way, the methodologies in project scheduling build upon each other. *Project scheduling requires us to follow some carefully laid-out steps, in order, for the schedule to take shape.* Just as a jigsaw puzzle will eventually yield a finished picture if we have followed the process correctly, the shape of the project's schedule will also come into direct focus when we learn the steps needed to bring it about.

9.1 PROJECT SCHEDULING

Project scheduling techniques lie at the heart of project planning and subsequent monitoring and control. Previous chapters have examined the development of vision and goals for the project, project screening activities, risk management practices, and project scope (including the Work Breakdown Structure). Project scheduling represents the conversion of project goals into an achievable methodology for their completion; it creates a timetable and reveals the network logic that relates project activities to each other in a coherent fashion. Because project management is predicated on completing a finite set of goals under a specified time frame, exactly how we develop the project's schedule is vitally important to success.

This chapter will examine a number of elements in project scheduling and demonstrate how to build the project plan from a simple set of identified project activities into a graphical set of sequential relationships between those tasks which, when performed, result in the completion of the project goals. Project **scheduling** has been defined by the Project Management Body of Knowledge as "an output of a schedule model that presents linked activities with planned dates, durations, milestones, and resources."[2] The term **linked activities** is important because it illustrates the scheduling goal. Project scheduling defines network logic for all **activities**; that is, **tasks** must either precede or follow other tasks from the beginning of the project to its completion.

Suppose you and your classroom team were given an assignment on leadership and were expected to turn in a paper and give a presentation at the end of the semester. It would first be necessary to break up the assignment into the discrete set of individual activities (Work Breakdown Structure) that would allow your team to finish the project. Perhaps you identified the following tasks needed to complete the assignment:

1. Identify topic
2. Research topic
3. Write first draft of paper
4. Edit and rewrite paper
5. Prepare class presentation
6. Complete final draft
7. Complete presentation
8. Hand in paper and present topic in class

Carefully defining all the steps necessary to complete the assignment is an important first step in project scheduling as it adds a sequential logic to the tasks and goes further in that it allows you to create a coherent project plan from start to finish. Suppose, to ensure the best use of your time and availability, you were to create a network of the activities listed above, that is, the most likely order in which they must occur to be done correctly. First, it would be necessary to determine a reasonable sequence. *Preceding activities* are those that must occur before others can be done. For example, it would be necessary to first identify the term paper topic before beginning to conduct research on it. Therefore, activity 1, *Identify topic,* is a preceding activity; and activity 2, *Research topic,* is referred to as a subsequent, or *successor, activity.*

Once you have identified a reasonable sequential logic for the network, you can construct a **network diagram**, which is a schematic display of the project's sequential activities and the logical relationships between them. Figure 9.2 shows two examples of a network diagram for your project. Note that in Option A, the easiest method for constructing a network diagram is to simply lay out all activities in serial order, starting with the first task and concluding with the final activity. This option, however, is usually not the most efficient one. It could be argued, for example, that it is not necessary that the whole project team be involved in each of the activities, requiring you to delay the start of activity 6, *Complete final draft* (F in Figure 9.2), until after activity 5, *Prepare class presentation.* Another choice might be to use the time better by having some members of the team begin work on the presentation while others are still completing the paper. Any of these options mean that you are now constructing a project network with two paths, or parallel streams of activities, some of which are going on simultaneously. This alternative network can be seen in Option B of Figure 9.2.

This simplified example illustrates the process of applying sequential logic to project tasks in order to construct an activity network. In creating a sense of timing for activities in addition to their functions, the activity network allows project teams to use a method for planning

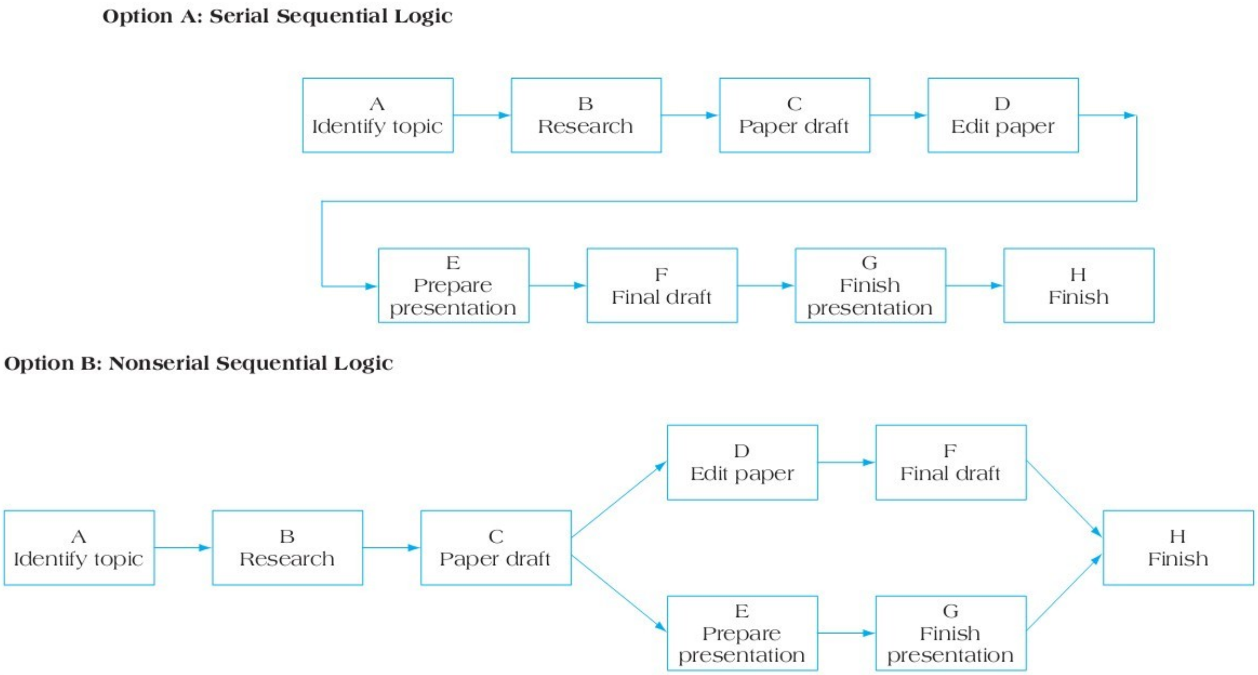


FIGURE 9.2 **Alternative Activity Networks for Term Paper Assignment**

and scheduling. There are several reasons why it is so important that project networks and scheduling be done well. Among the reasons are the following:[3]

- A network clearly illustrates the interdependence of all tasks and work packages. Doing something wrong earlier in the project has severe implications for downstream activities.
- Because a network illustrates this interrelationship among activities and project personnel, it facilitates communication flows. People are much more attuned to the work that went on before their involvement, and they develop a keener appreciation of the concerns of those who will take over at later points.
- A network helps with master scheduling of organizational resources because it shows times when various personnel must be fully committed to project activities. Without some sense of where the project fits into the overall organizational scheme, personnel may be assigned to multiple activities at a time when they are most needed on the project.
- A network identifies the critical activities and distinguishes them from the less critical. The network reveals the activities that absolutely must be completed on time to ensure that the overall project is delivered on time; in the process, activities that have some "wiggle room" are identified as well.
- Networks determine when you can expect projects to be completed.
- Dates on which various project activities must start and end in order to keep to the overall schedule are identified in a network.
- A network demonstrates which activities are dependent on which other activities. You then know the activities that need to be highly coordinated in order to ensure the smooth development of the project.

These are just some of the advantages of using activity networks for project scheduling.

9.2 KEY SCHEDULING TERMINOLOGY

Every profession has its unique jargon and terminology. In project scheduling, a number of specific terms are commonly employed and so need specific definitions. In many cases, their definitions are taken from the Project Management Institute's Body of Knowledge. Some concepts that you will see again and again throughout this chapter (and subsequent chapters) are listed here. You have already run across some of these terms in previous chapters.

Scope—The work content and products of a project or component of a project. Scope is fully described by naming all activities performed, the resources consumed, and the end products that result, including quality standards.

Work Breakdown Structure (WBS)—A task-oriented "family tree" of activities that organizes, defines, and graphically displays the total work to be accomplished in order to achieve the final objectives of a project. Each descending level represents an increasingly detailed definition of the project objective.

Work package—A deliverable at the lowest level of the Work Breakdown Structure; it is an element of work performed during the course of a project. A work package normally has an expected duration plus an expected cost. Other generic terms for project work include *task* or *activity*.

Project network diagram (PND)—Any schematic display of the logical relationships of project activities.

Path—A sequence of activities defined by the project network logic.

Event—A point when an activity is either started or completed. Often used in conjunction with AOA networks, events consume no resources and have no time to completion associated with them.

Node—One of the defining points of a network; a junction point joined to some or all of the others by dependency lines (paths).

Predecessors—Those activities that must be completed prior to initiation of a later activity in the network.

Successors—Activities that cannot be started until previous activities have been completed. These activities follow predecessor tasks.

Early start (ES) date—The earliest possible date on which the uncompleted portions of an activity (or the project) can start, based on the network logic and any schedule constraints. Early start dates can change as the project progresses and changes are made to the project plan.

Late start (LS) date—The latest possible date that an activity may begin without delaying a specified milestone (usually the project finish date).

Forward pass—Network calculations that determine the earliest start/earliest finish time (date) for each activity. The earliest start and finish dates are determined by working forward through each activity in the network.

Backward pass—Calculation of late finish times (dates) for all uncompleted network activities. The latest finish dates are determined by working backward through each activity.

Merge activity—An activity with two or more immediate predecessors (tasks flowing into it). Merge activities can be located by doing a forward pass through the network. The PMBoK refers to merge activities as "*path convergence*."

Burst activity—An activity with two or more immediate successor activities (tasks flowing out from it). Burst activities can be located by doing a backward pass through the network. The PMBoK refers to burst activities as "*path divergence*."

Float—The amount of time an activity may be delayed from its early start without delaying the finish of the project. Float is a mathematical calculation and can change as the project progresses and changes are made in the project plan. Also called *slack, total float,* and *path float.* In general, float is the difference between the late start date and the early start date (LS − ES) or between the late finish date and early finish date (LF − EF).

Critical path—The path through the project network with the longest duration. The critical path may change from time to time as activities are completed ahead of or behind schedule. Critical path activities are identified as having zero float in the project.

Critical Path Method (CPM)—A *network analysis* technique used to determine the amount of scheduling flexibility (the amount of float) on various logical network paths in the project schedule network, and to determine the minimum total project duration. It involves the calculation of early (forward scheduling) and late (backward scheduling) start and finish dates for each activity. Implicit in this technique is the assumption that whatever resources are required in any given time period will be available. Activities times are assumed to be known, or *deterministic*.

Resource-limited schedule—A project schedule whose start and finish dates reflect expected resource availability. The final project schedule should always be resource-limited.

Program Evaluation and Review Technique (PERT)—An event- and probability-based network analysis system generally used in projects where activities and their durations are difficult to define. PERT is often used in large programs where the projects involve numerous organizations at widely different locations.

The two most common methods for constructing activity networks involve **Activity-on-Arrow (AOA)** and **Activity-on-Node (AON)** logic. In the AOA method, the **arrow** represents the task, or activity, and the node signifies an event marker that suggests the completion of one activity and the potential to start the next. In AON methodology, the node represents an activity and the path arrows demonstrate the logical sequencing from node to node through the network. AOA approaches were most popular several decades ago and are still used to some extent in the construction industry, but with the rapid rise in computer-based scheduling programs, there is now a strong emphasis on AON methodology. Hence, in this chapter, we use AON examples and diagrams exclusively. Chapter 10 will discuss the rudiments of AOA network modeling.

9.3 DEVELOPING A NETWORK

Network diagramming is a logical, sequential process that requires you to consider the order in which activities should occur to schedule projects as efficiently as possible. There are two primary methods for developing activity networks, PERT and CPM. PERT, which stands for Program Evaluation and Review Technique, was developed in the late 1950s in collaboration between the U.S. Navy, Booz-Allen Hamilton, and Lockheed Corporation for the creation of the Polaris missile program. PERT originally was used in research and development (R&D), a field in which activity duration estimates can be difficult to make, and resulted from probability analysis. CPM, or Critical Path Method, was developed independently at the same time as PERT by DuPont, Inc. CPM, used commonly in the construction industry, differs from PERT primarily in the assumptions it makes about estimating activity durations. CPM assumes that durations are more deterministic; that is, they are easier to ascertain and can be assigned to activities with greater confidence. Further, CPM was designed to better link (and therefore control) project activity time and costs, particularly the time/cost trade-offs that lead to **crashing** decisions (speeding up the project). Crashing the project will be explained in more detail in Chapter 10. In practice, however, over the years the differences between PERT and CPM have blurred to the point where it is now common to simply refer to these networking techniques as PERT/CPM.[4]

Prior to constructing an activity network, there are some simple rules of thumb you need to become familiar with as you develop the network diagram. These rules are helpful in understanding the logic of activity networks.[5]

1. Some determination of activity precedence ordering must be done prior to creating the network. That is, all activities must be logically linked to each other—those that precede others, as well as successor activities (those that must follow others).
2. Network diagrams usually flow from left to right.
3. An activity cannot begin until all preceding connected activities have been completed.
4. Arrows on networks indicate precedence and logical flow. Arrows can cross over each other, although it is helpful for clarity's sake to limit this effect when possible.
5. Each activity should have a unique identifier associated with it (number, letter, code, etc.). For simplicity, these identifiers should occur in ascending order; each one should be larger than the identifiers of preceding activities.
6. Looping, or recycling through activities, is not permitted.
7. Although not required, it is common to start a project from a single beginning node, even in the case when multiple start points are possible. A single node point also is typically used as a project end indicator.

With these simple rules of thumb firmly in mind, you can begin to uncover some of the basic principles of establishing a network diagram. Remember that AON methodology represents all activities within the network as nodes. Arrows are used only to indicate the sequential flow of activities from the start of the project to its conclusion

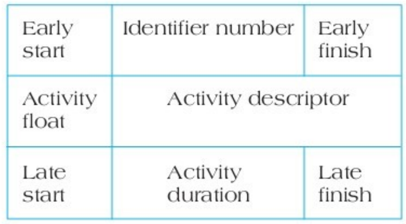


FIGURE 9.3 **Labels for Activity Node**

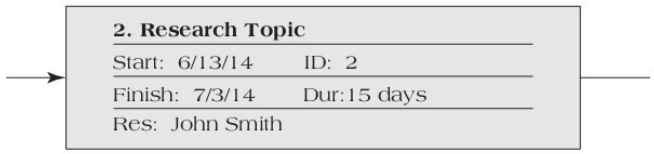


FIGURE 9.4 **Activity Node Labels Using MS Project 2013**

Source: MS Project 2013, Microsoft Corporation.

Labeling Nodes

Nodes representing project activities should be clearly labeled with a number of different pieces of information. It is helpful if the nodes at least contain the following data: (1) identifier, (2) descriptive label, (3) activity duration, (4) early start time, (5) early finish time, (6) late start time, (7) late finish time, and (8) activity float. Figure 9.3 shows the labeling for a node with each piece of information assigned to a location within the activity box. The arrangement selected for this node was arbitrary; there is no accepted standard for labeling activity nodes. For example, the node shown in Figure 9.4 was derived from a standard Microsoft Project 2013 output file. Note that in this example, the activity start and finish dates are shown, as well as the resource person responsible for the activity's completion.

Complete labels on activity nodes make it easier to use the network to perform additional calculations such as identifying critical path, activity float (or slack), total project duration, and so on. When constructing network diagrams during the early development of the project, all necessary information about the activity can be retrieved quickly as long as nodes are fully labeled.

Serial Activities

Serial activities are those that flow from one to the next, in sequence. Following the logic of Figure 9.5, we cannot begin work on activity B until activity A has been completed. Activity C cannot begin until both activities A and B are finished. Serial activity networks are the simplest in that they create only linkages of activity sequencing. In many cases, serial networks are appropriate representations of the project activities. Figure 9.5 demonstrates how, in the earlier example of preparing for a term paper and presentation, several activities must necessarily be linked serially. Identifying the topic, conducting research, and writing the first draft are activities that must link in series, because subsequent activities cannot begin until the previous (predecessor) ones have been completed.

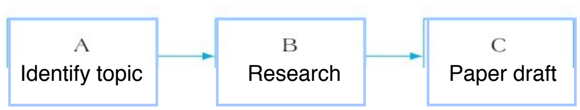
Network logic suggests that:

Activity A can begin immediately.
Activity B cannot begin until activity A is completed.
Activity C cannot begin until both activities A and B are completed.

Concurrent Activities

In many circumstances, it is possible to begin work on more than one activity simultaneously, assuming that we have the resources available for both. Figure 9.6 provides an example of how

FIGURE 9.5 **Project Activities Linked in Series**



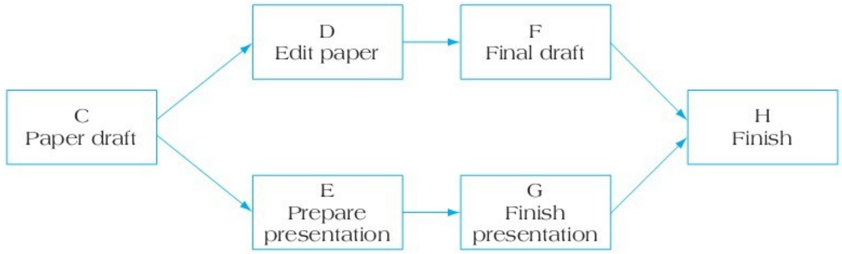


FIGURE 9.6 **Activities Linked in Parallel (Concurrent)**

concurrent or parallel project paths are represented in an activity network. When the nature of the work allows for more than one activity to be accomplished at the same time, these activities are called **concurrent**, and parallel project activity paths are constructed through the network. In order to successfully operate concurrent activities, the project must be staffed with sufficient human resources to support all simultaneous activities. This is a critical issue, because a network cannot be created without giving thought to the resource requirements needed to support it.

Network logic suggests that:

Activities D and E can begin following the completion of activity C.
Activity F can begin following the completion of activity D and is independent of activity E.
Activity G can begin following the completion of activity E and is independent of activity D.
Activity H can begin following the completion of both activities F and G.

Merge Activities

Merge activities are those with two or more immediate predecessors. Figure 9.7 is a partial network diagram that shows how merge activities are expressed graphically. Merge activities often are critical junction points, places where two or more parallel project paths converge within the overall network. Figure 9.7 demonstrates the logic of a merge activity: You cannot begin activity D until all predecessor activities, A, B, and C, have been completed. The start of the merge activity is subject to the completion of the longest prior activity. For example, suppose that activities A, B, and C all start on the same day. Activity A has a duration of 3 days, activity B's duration is 5 days, and activity C has a duration of 7 days. The earliest activity D, the merge point, can start is on day 7, following completion of all three predecessor activities.

Network logic suggests that:

Activity D can only begin following the completion of activities A, B, and C.

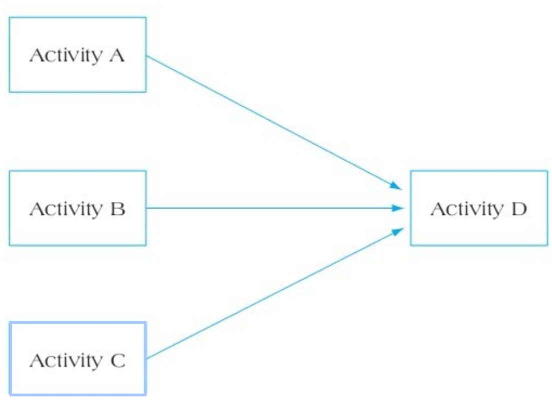


FIGURE 9.7 **Merge Activity**

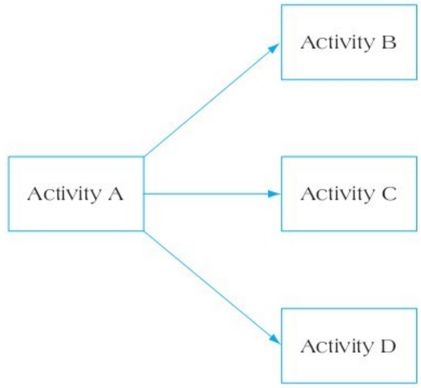


FIGURE 9.8 Burst Activity

Burst Activities

Burst activities are those with two or more immediate successor activities. Figure 9.8 graphically depicts a burst task, with activities B, C, and D scheduled to follow the completion of activity A. All three successors can only be undertaken upon the completion of activity A. Unlike merge activities, in which the successor is dependent upon completion of the longest predecessor activity before it can begin, all immediate successors can begin simultaneously upon completion of the burst activity.

Network logic suggests that:

Activities B, C, and D can only begin following the completion of activity A.

EXAMPLE 9.1

Let's begin constructing a basic activity network. Table 9.1 identifies eight activities and their predecessors in a simple example project. Once we have determined the tasks necessary to accomplish the project, it is important to begin linking those tasks to each other. In effect, we are taking the project tasks in the Work Breakdown Structure and adding a project chronology.

Once the network activity table has been developed and the predecessors identified, we can begin the process of network construction. The first activity (A) shows no predecessors; it is the starting point in the network and placed to the far left of our diagram. Next, activities B and C both identify activity A as their predecessor. We can place them on the network as well. Activity D lists both activities B and C as predecessors. Figure 9.9 gives a partial network diagram based on the information we have compiled to this point. Note that, based on our definitions, activity A is a burst activity and activity D is a merge activity.

We can continue to create the network iteratively as we add additional activity nodes to the diagram. Figure 9.10 shows the final activity network. Referring back to an earlier point,

TABLE 9.1 Information for Network Construction

Name: Project Delta

Activity	Description	Predecessors
A	Contract signing	None
B	Questionnaire design	A
C	Target market ID	A
D	Survey sample	B, C
E	Develop presentation	B
F	Analyze results	D
G	Demographic analysis	C
H	Presentation to client	E, F, G

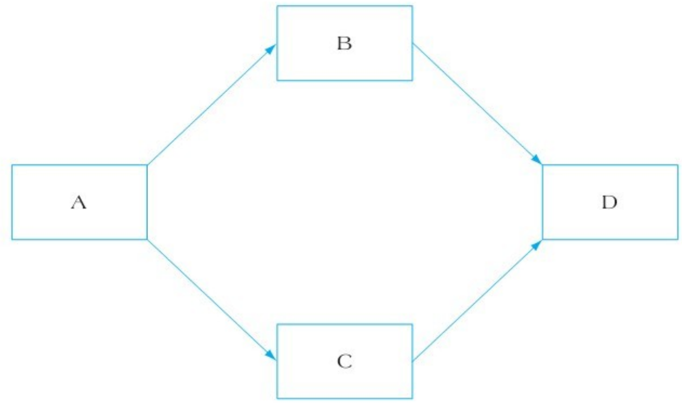


FIGURE 9.9 Partial Activity Network Based on Project Delta

note that this network begins with a single node point (activity A) and concludes with a single point (activity H). The merge activities associated with this network include activities D (with activities B and C merging at this node) and H (with activities E, F, and G merging at this node). Activities A, B, and C are burst activities. Recall that burst activities are defined as those with two or more immediate successors in the network. Activity A has the successor tasks B and C, activity B has tasks D and E following it, and activity C has two successors (D and G).

If we employed Microsoft Project 2013 to create the network diagram, we would first enter each of the activities into the template shown in Figure 9.11. Note that for this example, we are not assigning any durations to the activities, so the default is set at 1 day for each activity.

The next step in using MS Project to create a network is to identify the predecessor activities at each step in the project. In Figure 9.12, we begin to build the network by specifying each predecessor and successor in the network. Double-clicking the mouse on an activity will bring up a Task Information window (shown in Figure 9.12). In that window, we can specify the task or tasks that are predecessors of our current activity. For activity B (questionnaire design), we have specified a single predecessor (contract signing).

Once we have added each task in turn, the project network is completed. MS Project can be used to generate the final network, as shown in Figure 9.13. Note that each activity is still labeled as needing only 1 day for completion. In the next section of this chapter, we begin to consider the manner in which individual activity durations can be determined.

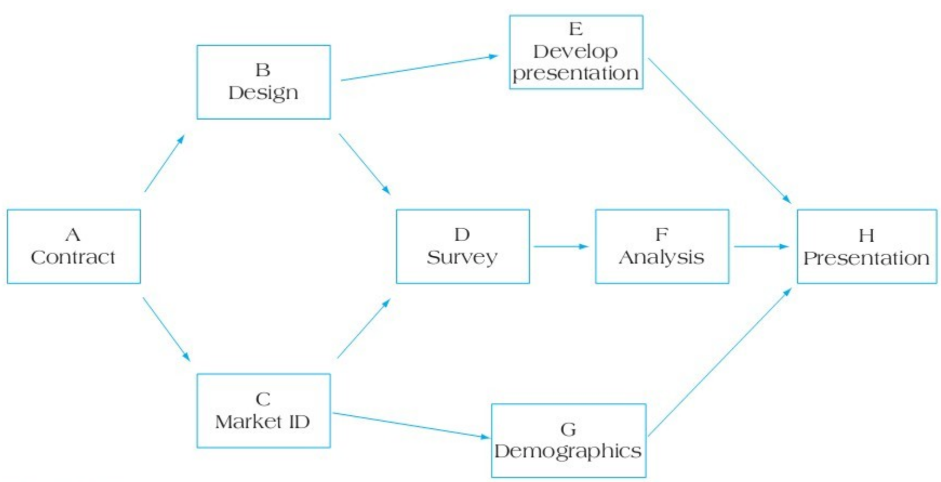


FIGURE 9.10 Complete Activity Network for Project Delta

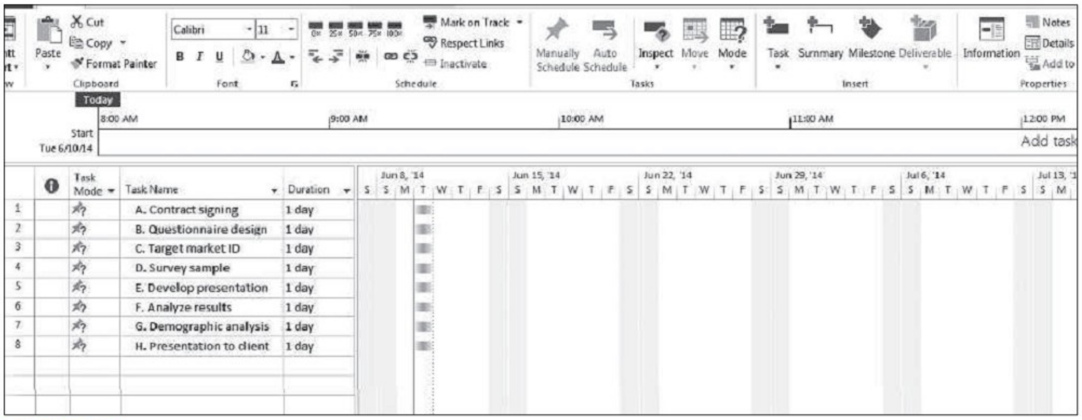


FIGURE 9.11 **Developing the Activity Network Using MS Project 2013**

Source: MS Project 2013, Microsoft Corporation.

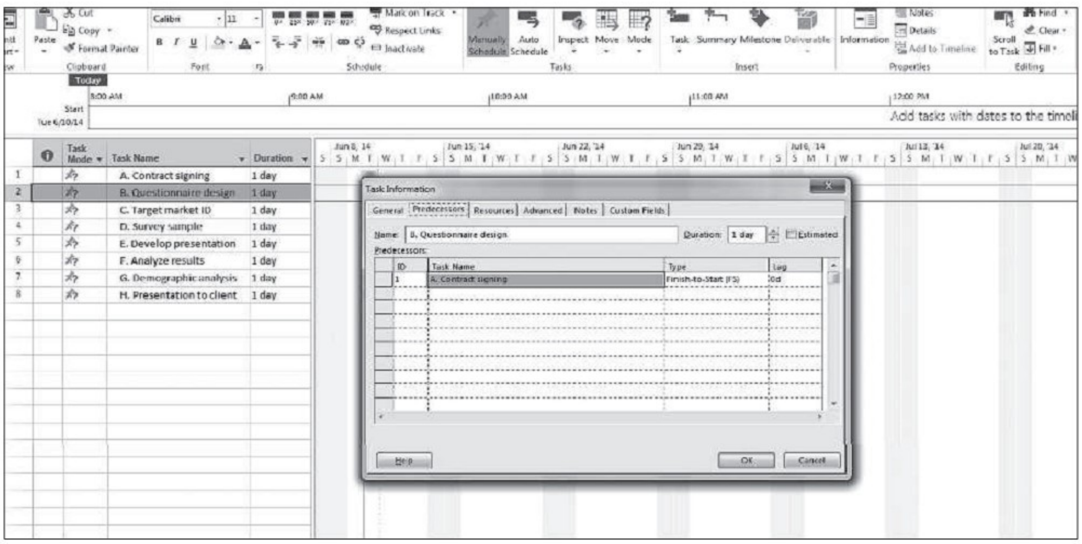


FIGURE 9.12 **Task Information Window Used to Specify Predecessors for Activity Networks**

Source: MS Project 2013, Microsoft Corporation.

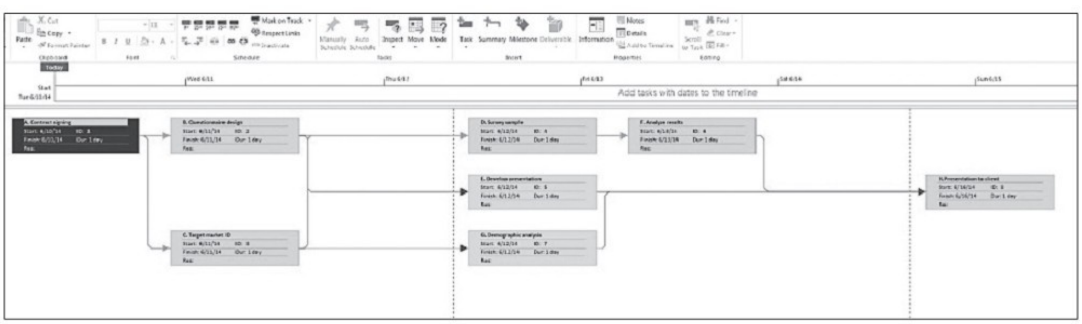


FIGURE 9.13 **The Completed MS Project 2010 Network Diagram**

Source: MS Project 2013, Microsoft Corporation.

9.4 DURATION ESTIMATION

The next step in building the network is to estimate activity **durations** for each step in the project. The first point to remember is that these estimates are based on what is assumed to be normal working methods during normal business or working hours. Second, although factors such as past experience or familiarity with the work will influence the accuracy of these estimates, activity durations are always somewhat uncertain. Third, time frames for task estimates can vary from several hours for short projects to days and weeks for longer projects.

Activity durations can be estimated in a number of different ways, including:[6]

- ***Experience.*** In cases where the organization has previously done similar work, we can use history as a guide. This approach is relatively easy; we simply call upon past examples of similar projects and use them as a baseline. The main drawback to this approach is that it assumes what worked in the past will continue to work today. Projects are affected by external events that are unique to their own time. Therefore, in using experience, we must be aware of the potential for using distorted or outdated information.
- ***Expert opinion.*** At times we may be told to contact a past project manager or expert in a particular area to get accurate information on activity estimates. Intuitively this approach would seem to be useful—if you want to know something, go to an expert. Yet "experts" are considered experts precisely because they know the easiest avenues, best contacts, and fastest processes to complete tasks. Would an expert's estimate of completion time be valid for nonexperts doing the same activity? The answer is not absolute, but the question suggests that we use caution in our application of expert opinion.
- ***Mathematical derivation.*** Another approach offers a more objective alternative to activity duration estimation and sidesteps many of the problems that can be found in more subjective methods. This method consists of developing duration probability based on a reasoned analysis of best-case, most likely case, and worst-case scenarios.

There are two primary means for developing duration estimates. We discussed earlier in this chapter (Section 9.3) the idea that the simplest approach is to assume a deterministic model for activity durations. *Deterministic estimation* means that activity durations are fairly predictable; that is, they do not consider variation in the activity completion time. So, for example, when developing a construction project, we can call upon years of experience to know that the time it takes to dig the foundation and pour concrete "footers" for a 2,500-square-foot residential construction will be 10 hours. This is an example of a predictable, or deterministic, time estimate. On the other hand, for many project activities, we can only make educated estimates of their likely duration, based partially on past experience, but also requiring us to take into consideration the likelihood of variation in how long the activity may take to complete. Developing a new software procedure using the latest generation of programming code (one which our company's programmers are still learning) can be a difficult project with which to make activity duration estimates. It is for this reason that mathematical procedures have been created to help determine activity times.

In order to understand how to use mathematical derivation to determine expected activity times, we need to consider the basics of probability distributions. Probability suggests that the amount of time an activity is likely to take can rarely be positively determined; rather, it is found as the result of sampling a range of likelihoods, or probabilities, of the event occurring. These likelihoods range from 0 (no probability) to 1 (complete probability). In order to derive a reasonable probabilistic estimate for an activity's duration, we need to identify three values: (1) the activity's most likely duration, (2) the activity's most pessimistic duration, and (3) the activity's most optimistic duration. The most likely duration is determined to be the length of time expected to complete an activity assuming the development of that activity proceeds normally. Pessimistic duration is the expected length of time needed to develop the activity under the assumption that everything will go badly (Murphy's Law). Finally, optimistic duration is estimated under the assumption that the development process will proceed extremely well.

For these time estimates, we can use probability distributions that are either symmetrical (the normal distribution) or asymmetrical (the beta distribution). A normal distribution implies that the probability of an event taking the most likely time is one that is centered on the mean of the distribution (see Figure 9.14). Because pessimistic and optimistic values are estimated at the 95% confidence level from either end of the distribution, they will cancel each other out, leaving the mean value as the expected duration time for the activity.

In real life it is extremely rare to find examples in which optimistic and pessimistic durations are symmetrical to each other about the mean. In project management, it is more common to see probability distributions that are asymmetrical; these are referred to as **beta distributions**. The asymmetry of the probability distribution suggests we recognize that certain events are less likelv
mean while its pessimistic time may be as much as three or four standard deviations away. To illustrate, suppose that we began construction on a highway bridge and wished to estimate the length

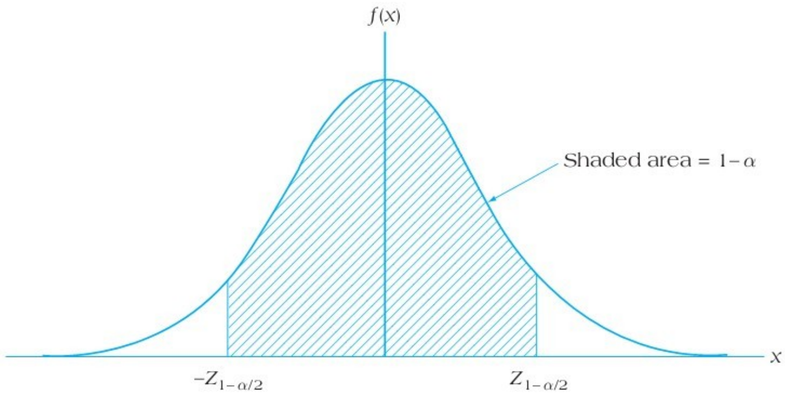


FIGURE 9.14 **Symmetrical (Normal) Distribution for Activity Duration Estimation**

of time (duration) it would take to place the steel girders needed to frame the bridge. We expect that the duration for the framing task will take six days; however, a number of factors could change that duration estimate. We could, for example, experience uncommonly good weather and have no technical delays, allowing us to finish the framing work in only four days. On the other hand, we could have terrible weather, experience delivery delays for needed materials, and lose time in labor disputes, all leading to a pessimistic estimate of 14 days. This example demonstrates the asymmetrical nature of duration estimates; while our most likely duration is 6 days, the range can vary from 4 to 14 days to complete the task.

The optimistic and pessimistic duration values essentially serve as upper and lower bounds for the distribution range. Figure 9.15 illustrates a beta distribution with the values m (most likely duration), a (most optimistic duration), and b (most pessimistic duration) identified.

Two assumptions are used to convert the values of m, a, and b into estimates of the expected time (TE) and variance (s^2) of the duration for the activity. One important assumption is that s, the standard deviation of the duration required to complete the task, equals one-sixth of the range for reasonably possible time requirements. The **variance** for an activity duration estimate is given by the formula:

$$s^2 = [(b - a)/6]^2$$

The logic for this assumption is based on the understanding that to achieve a probability distribution with a 99% confidence interval, observations should lie within three standard deviations of the mean in either direction. A spread of six standard deviations from tail to tail in the probability distribution, then, accounts for 99.7% of the possible activity duration alternatives.

Because optimistic and pessimistic times are not symmetrical about the mean, the second assumption refers to the shape of the probability distribution. Again, the beta, or asymmetrical, distribution better represents the distribution of possible alternative expected duration times (TE) for estimating activities. The beta distribution suggests that the calculation for deriving TE is shown as:

$$\text{TE} = (a + 4m + b)/6$$

where

TE = estimated time for activity
a = most optimistic time to complete the activity
m = most likely time to complete the activity, (the mode of the distribution)
b = most pessimistic time to complete the activity

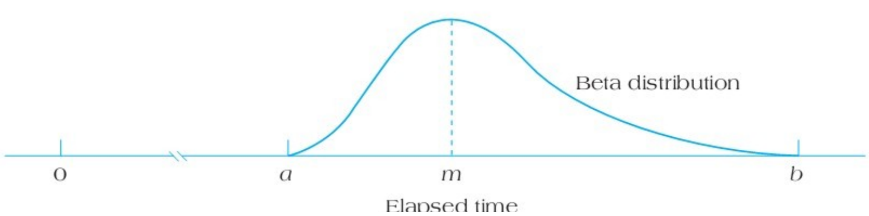


Figure 9.15 Asymmetrical (Beta) Distribution for Activity Duration Estimation

TABLE 9.2 Activity Duration Estimates for Project Delta

Name: Project Delta
Durations are listed in weeks

Activity	Description	Optimistic	Likely	Pessimistic
A	Contract signing	3	4	11
B	Questionnaire design	2	5	8
C	Target market ID	3	6	9
D	Survey sample	8	12	20
E	Develop presentation	3	5	12
F	Analyze results	2	4	7
G	Demographic analysis	6	9	14
H	Presentation to client	1	2	4

In this calculation, the midpoint between the pessimistic and optimistic values is the weighted arithmetic mean of the mode and midrange, representing two-thirds of the overall weighting for the calculated expected time. The additional weighting is intended to highlight the clustering of expected values around the distribution mean, regardless of the length of both pessimistic and optimistic tails (total distribution standard deviation).

How do we put together all of these assumptions to perform an accurate activity duration estimation? The next step is to construct an activity duration estimate table (see Table 9.2). For simplicity, all numbers shown are in weeks.

Table 9.2 demonstrates the most likely times for each activity based on a reasonably accurate assessment of how long a task *should* take, *could* take if everything went well, and *would* take if everything went poorly. If we assign the value a to the most optimistic duration estimate, the project manager must assign a value to this activity such that the actual amount of time needed to complete the activity will be a or greater 99% of the time. Conversely, in assigning a value for the most pessimistic duration, b, the project manager should estimate the duration of the activity to have a 99% likelihood that it will take b or less amount of time.

The standard formula for estimating expected activity duration times is based on the weighting ratio of $1 \times$ optimistic, $4 \times$ likely, and $1 \times$ pessimistic. Researchers and practitioners alike, however, have found that this ratio is best viewed as a heuristic whose basic assumptions are affected by a project's unique circumstances. One argument holds that the above ratio is far too optimistic and does not take into consideration the negative impact created when the worst-case or pessimistic estimate proves accurate. Further, given the inherent uncertainty in many projects, significant levels of risk must be accounted for in all probabilistic estimates of duration.

Extensive research into the topic of improving the accuracy of activity duration estimation has not led to definitive results. Modeling techniques such as Monte Carlo simulation and linear and nonlinear programming algorithms generally have demonstrated that the degree of uncertainty in task durations can have a significant impact on the optimum method for duration estimation. Because uncertainty is so common in activity estimation, more than one activity estimate may be reasonably held. The goal is to achieve a **confidence interval** that provides the highest reasonable probability of being accurate. Probability estimation using 99% confidence intervals represents a degree of confidence few project managers would be willing to demonstrate, according to Meredith and Mantel.[7] Consequently, when the confidence interval level assumption is relaxed to, for example, 90%, the variance calculations and estimates of duration must be modified accordingly. Although the debate is likely to continue, an estimation formula of 1:4:1 (optimistic:likely:pessimistic)/6 is commonly accepted.

Using this ratio as a tool, it is now possible to calculate expected activity duration times for each of the tasks identified in Table 9.2. Table 9.3 shows the calculated times for each activity, based on the assumption of a beta distribution.

TABLE 9.3 Estimated Project Activity Times Using Beta Distribution

Name: Project Delta		
Durations are listed in weeks		
Activity	**Description**	**Beta (1:4:1 ratio)/6**
A	Contract signing	5
B	Questionnaire design	5
C	Target market ID	6
D	Survey sample	12.7
E	Develop presentation	5.8
F	Analyze results	4.2
G	Demographic analysis	9.3
H	Presentation to client	2.2

Creating the project network and calculating activity durations are the first two key steps in developing the project schedule. The next stage is to combine these two pieces of information in order to create the project's critical path diagram.

9.5 CONSTRUCTING THE CRITICAL PATH

The next step is to link activity duration estimates and begin construction of the critical path. Critical path calculations link activity durations to the preconstructed project activity network. This point is important: The project network is first developed using activity precedence logic, *then*, following task duration estimates, these values are applied in a structured process to each activity to determine overall project length. In addition to allowing us to determine how long the project is going to take, applying time estimates to the network lets us discover activity float (which activities can be delayed and which cannot), the latest and earliest times each activity can be started or must be completed, and the latest and earliest times each activity can be completed.

Calculating the Network

The process for developing the network with time estimates is fairly straightforward. Once the activity network and duration estimates are in place, the actual network calculation computations can proceed. Look again at the network in Figure 9.10 and the duration estimates given in Table 9.3 that assume a beta distribution. In this example, the time estimates are rounded to the nearest whole integer. The activity information is summarized in Table 9.4.

TABLE 9.4 Project Information

Project Delta			
Activity	**Description**	**Predecessors**	**Estimated Duration**
A	Contract signing	None	5
B	Questionnaire design	A	5
C	Target market ID	A	6
D	Survey sample	B, C	13
E	Develop presentation	B	6
F	Analyze results	D	4
G	Demographic analysis	C	9
H	Presentation to client	E, F, G	2

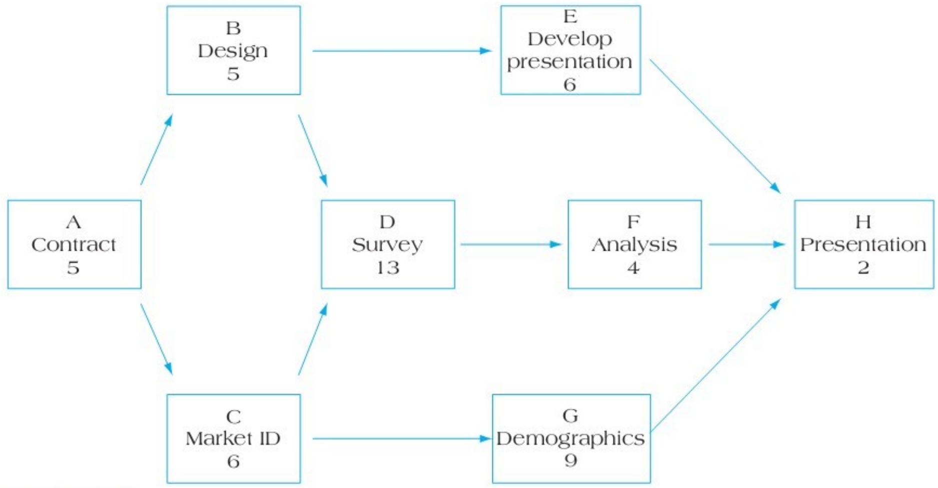


FIGURE 9.16 **Partial Project Activity Network with Task Durations**

The methodology for using this information to create a critical path requires two steps: a *forward pass* through the network from the first activity to the last and a *backward pass* through the network from the final activity to the beginning. The forward pass is an additive process that calculates the earliest times an activity can begin and end. Once we have completed the forward pass, we will know how long the overall project is expected to take. The backward pass is a subtractive process that gives us information on when the latest activities can begin and end. Once both the forward and backward passes have been completed, we will also be able to determine individual activity float and, finally, the project's critical path.

After labeling the network with the activity durations, we begin to determine the various paths through the network. Figure 9.16 shows a partial activity network with durations labeled for each of the eight project activities. Each path is discovered by assessing all possible sequences of precedence activities from the beginning node to the end. Here, we can identify four separate paths, labeled:

Path One: A − B − E − H

Path Two: A − B − D − F − H

Path Three: A − C − D − F − H

Path Four: A − C − G − H

Since we now know the activity times for each task, we can also identify the critical path. The *critical path* is defined as the "series of interdependent activities of a project, connected end-to-end, which determines the shortest total length of the project."[8] The shortest total length of time needed to complete a project is determined by the *longest path* through the network. The length of the four paths listed above can be derived simply by adding their individual activity durations together. Hence,

Path One: A − B − E − H = 18 weeks

Path Two: A − B − D − F − H = 29 weeks

Path Three: A − C − D − F − H = 30 weeks

Path Four: A − C − G − H = 22 weeks

Path Three, which links the activities A – C – D – F – H, is scheduled for duration of 30 weeks and is the critical path for this activity. In practical terms, this path has no float, or slack time, associated with it.

The Forward Pass

We can now begin adding more information to the network by conducting the forward pass to determine the earliest times each activity can begin and the earliest it can be completed. The process is iterative: each step builds on the information contained in the node immediately preceding

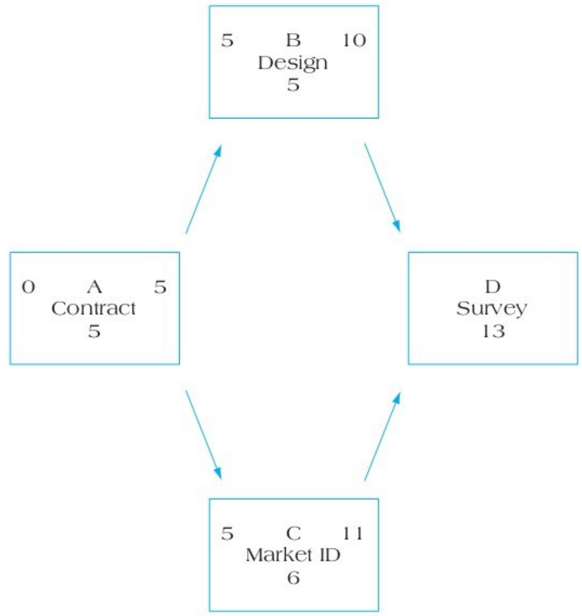


FIGURE 9.17 Partial Activity Network with Merge Point at Activity D

it in the network. The beginning activity, contract signing, can be started at time 0 (immediately). Therefore, the earliest that activity A can be completed is on day 5. Early finish for any activity (EF) is found by taking its early start (ES) time and adding its activity duration (ES + Dur = EF). Therefore, activity B (questionnaire design) has an activity early start time of 5. This value corresponds to the early finish of the activity immediately preceding it in the network. Likewise, activity C, which is also dependent upon the completion of activity A to start, has an early start of 5. The early finish for activity B, calculated by (ES + Dur = EF), is 5 + 5, or 10. The early finish for activity C is found by 5 + 6 = 11. Figure 9.17 shows the process for developing the forward pass through the activity network.

The first challenge occurs at activity D, the merge point for activities B and C. Activity B has an early finish (EF) time of 10 weeks; however, activity C has an EF of 11 weeks. What should be the activity early start (ES) for activity D?

In order to answer this question, it is helpful to review the rules that govern the use of forward pass methodology. Principally, there are three steps for applying the forward pass:

1. Add all activity times along each path as we move through the network (ES + Dur = EF).
2. Carry the EF time to the activity nodes immediately succeeding the recently completed node. That EF becomes the ES of the next node, unless the succeeding node is a merge point.
3. At a merge point, the largest preceding EF becomes the ES for that node.

Applying these rules, at activity D, a merge point, we have the option of applying either an EF of 10 (activity B) or of 11 (activity C) as our new ES. Because activity C's early finish is larger, we would select the ES value of 11 for this node. The logic for this rule regarding merge points is important: Remember that early start is defined as the earliest an activity can begin. When two or more immediate predecessors have varying EF times, *the earliest the successor can begin is when all preceding activities have been completed.* Thus, we can determine that it would be impossible for activity D to begin at week 10 because one of its predecessors (activity C) would not have been finished by that point.

If we continue applying the forward pass to the network, we can work in a straightforward manner until we reach the final node, activity H, which is also a merge point. Activity H has three immediate predecessors, activities E, F, and G. The EF for activity E is 16, the EF for activity F is 28, and the EF for activity G is 20. Therefore, the ES for activity H must be the largest EF, or 28. The final length of the project is 30 weeks. Figure 9.18 illustrates the overall network with all early start and early finish dates indicated.

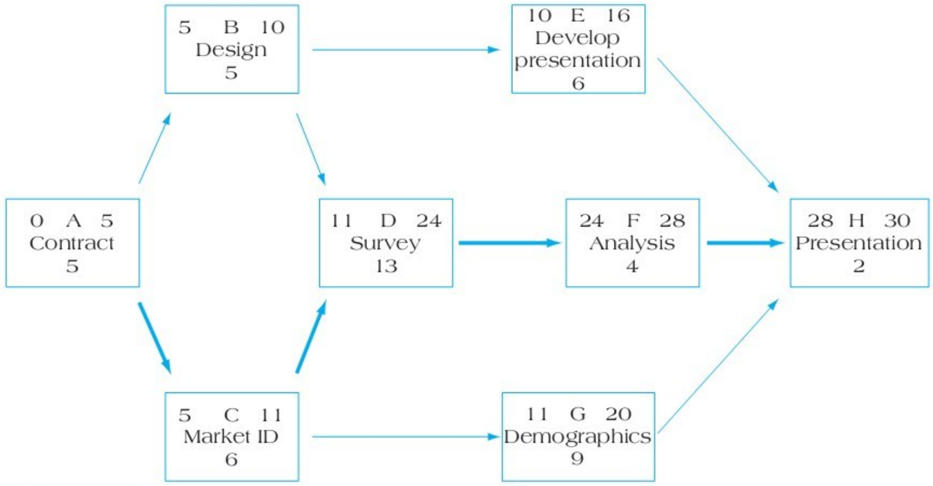


FIGURE 9.18 **Activity Network with Forward Pass**

The Backward Pass

We now are able to determine the overall length of the project, as well as each activity's early start and early finish times. When we take the next step of performing the backward pass through the network, we will be able to ascertain the project's critical path and the total float time of each project activity. The backward pass is an iterative process, just as the forward pass is. The difference here is that we begin at the end of the network, with the final node. The goal of the backward pass is to determine each activity's late start (LS) and late finish (LF) times. LS and LF are determined through a subtractive methodology.

In Figure 9.19, we begin the backward pass with the node representing activity H (presentation). The first value we can fill out in the node is the late finish (LF) value for the project. This value is the same as the early finish (30 weeks). For the final node in a project network, the EF = LF. Once we have identified the LF of 30 weeks, the LS for activity H is the difference between the LF and the activity's duration; in this case, $30 - 2 = 28$. The formula for determining LS is LF − Dur = LS. Thus, the LS for activity H is 28 and the LF is 30. These values are shown in the bottom of the node, with the LS in the bottom left corner and the LF in the bottom right corner. In order to determine the LF for the next three activities that are linked to activity H (activities E, F, and G), we carry the LS value of activity H backward to these nodes. Therefore, activities E, F, and G will each have 28 as their LF value.

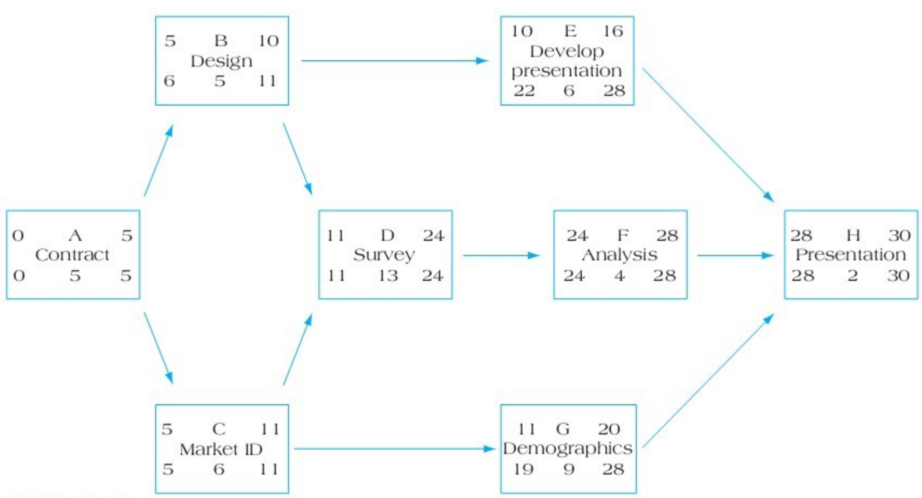


FIGURE 9.19 **Activity Network with Backward Pass**

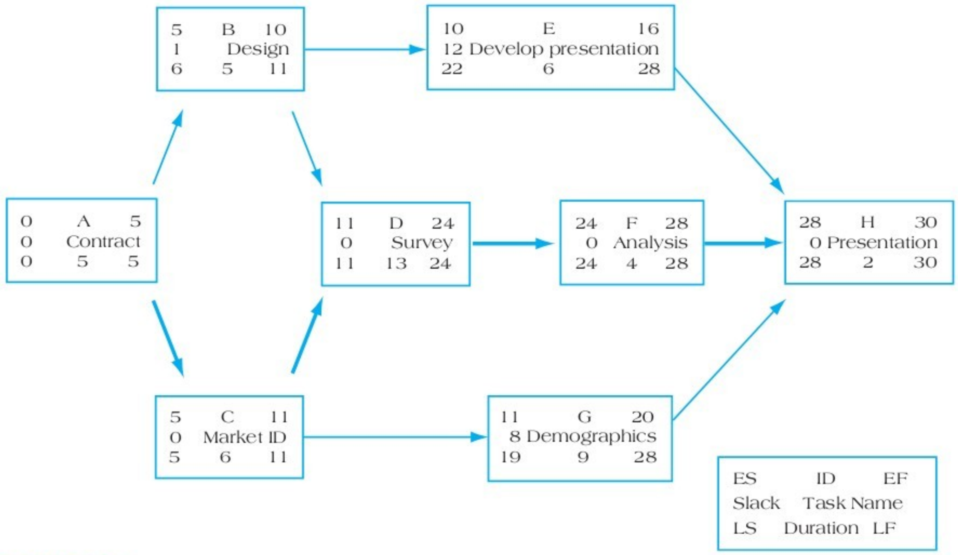


FIGURE 9.20 Project Network with Activity Slack and Critical Path

Note: Critical path is indicated with bold arrows.

Again, we subtract the durations from the LF values of each of the activities. The process continues to proceed backward, from right to left, through the network. However, just as in the forward pass we came upon a problem at merge points (activities D and H), we find ourselves in similar difficulty at the burst points—activities A, B, and C. At these three nodes, more than one preceding activity arrow converges, suggesting that there are multiple options for choosing the correct LF value. Burst activities, as we defined them, are those with two or more immediate successor activities. With activity B, both activities D and E are successors. For activity D, the LS = 11, and for activity E, the LS = 22. Which LS value should be selected as the LF for these burst activities?

To answer this question, we need to review the rules for the backward pass.

1. Subtract activity times along each path as you move through the network (LF − Dur = LS).
2. Carry back the LS time to the activity nodes immediately preceding the successor node. That LS becomes the LF of the next node, unless the preceding node is a burst point.
3. In the case of a burst point, the smallest succeeding LS becomes the LF for that node.

The correct choice for LF for activity B is 11 weeks, based on activity D. The correct choice for activity C, either 11 or 19 weeks from the network diagram, is 11 weeks. Finally, the LS for activity B is 6 weeks and it is 5 weeks for activity C; therefore, the LF for activity A is 5 weeks. Once we have labeled each node with its LS and LF values, the backward pass through the network is completed.

We can now determine the float, or **slack**, for each activity as well as the overall critical path. Again, float informs us of the amount of time an activity can be delayed and still not delay the overall project. Activity float is found through using one of two equations: LF − EF = Float or LS − ES = Float. Consider activity E with 12 weeks of float. Assume the worst-case scenario, in which the activity is unexpectedly delayed 10 weeks, starting on week 20 instead of the planned week 10. What are the implications of this delay on the overall project? None. With 12 weeks of float for activity E, a delay of 10 weeks will not affect the overall length of the project or delay its completion. What would happen if the activity were delayed by 14 weeks? The ES, instead of 10, is now 24. Adding activity duration (6 weeks), the new EF is 30. Take a look at the network shown in Figure 9.20 to see the impact of this delay. Because activity H is a merge point for activities E, F, and G, the largest EF value is the ES for the final node. The new largest EF is 30 in activity E. Therefore, the new node EF = ES + Dur, or 30 + 2 = 32. The effect of overusing available slack delays the project by 2 weeks.

One other important point to remember about activity float is that *it is determined as a result of performing the forward and backward passes through the network.* Until we have done the calculations for ES, EF, LS, and LF, we cannot be certain which activities have float associated with them and which do not. Using this information to determine the project critical path suggests that *the critical path is the network path with no activity slack associated with it.* In our project, we can determine the critical path by linking the nodes with no float: A – C – D – F – H. The only time this rule is violated is when an arbitrary value has been used for the project LF; for example, suppose that a critical deadline date is inserted at the end of the network as the LF. Regardless of how many days the project is calculated to take based on the forward pass calculation, if a deadline is substituted for the latest possible date to complete the project (LF), there is going to be some negative float associated with the project. *Negative float* refers to delays in which we have used up all available safety, or float, and are now facing project delays. For example, if top management unilaterally sets a date that allows the project only 28 weeks to the LF, the project critical path will start with 2 weeks of negative slack. It is often better to resolve problems of imposed completion dates by paring down activity estimates rather than beginning the project with some stored negative float.

We can also determine path float; that is, the linkage of each node within a noncritical path. The path A – B – E – H has a total of 13 weeks of float; however, it may be impossible to "borrow" against the float of later activities if the result is to conflict with the critical path. Although there are 13 weeks of float for the path, activity B cannot consume more than one week of the total float before becoming part of the critical path. This is because B is a predecessor activity to activity D, which is on the critical path. Using more than one week of extra float time to complete activity B will result in delaying the ES for critical activity D and thereby lengthening the project's critical path.

Probability of Project Completion

Calculating the critical path in our example shows us that the expected completion of Project Delta was 30 weeks, but remember that our original time estimates for each activity were probabilistic, based on the beta distribution. This implies that there is the potential for variance (perhaps *serious* variance) in the overall estimate for project duration. Variations in activities on the critical path can affect the overall project completion time and possibly delay it significantly. As a result, it is important to consider the manner in which we calculate and make use of activity duration variances. Recall that the formula for variance in activity durations is:

$$s^2 = [(b - a)/6]^2, \text{ where } b \text{ is the most pessimistic time and } a \text{ is the most optimistic}$$

Determining the individual activity variances is straightforward. As an example, let's refer back to Table 9.3 to find the variance for activity A (contract signing). Since we know the most optimistic and pessimistic times for this task (3 and 11 days, respectively), we calculate its variance as:

$$\text{Activity A}: [(11 - 3)/6]^2 = (8/6)^2 = 64/36, \text{ or } 1.78 \text{ weeks}$$

This information is important for project managers because we want to know not just likely times for activities but also how much confidence we can place in these estimates; thus, for our project's activity A, we can see that although it is most likely that it will finish in 5 weeks, there is a considerable amount of variance in that estimate (nearly 2 weeks). It is also possible to use this information to calculate the expected variance and standard deviation for all activities in our Project Delta, as Table 9.5 demonstrates.

TABLE 9.5 Expected Activity Durations and Variances for Project Delta

Activity	Optimistic (a)	Most Likely (m)	Pessimistic (b)	Expected Time	Variance $[(b - a)/6]^2$
A	3	4	11	5	$[(11 - 3)/6]^2 = 64/36 = 1.78$
B	2	5	8	5	$[(8 - 2)/6]^2 = 36/36 = 1.00$
C	3	6	9	6	$[(9 - 3)/6]^2 = 36/36 = 1.00$
D	8	12	20	12.7	$[(20 - 8)/6]^2 = 144/36 = 4.00$
E	3	5	12	5.8	$[(12 - 3)/6]^2 = 81/36 = 2.25$
F	2	4	7	4.2	$[(7 - 2)/6]^2 = 25/36 = 0.69$
G	6	9	14	9.3	$[(14 - 6)/6]^2 = 64/36 = 1.78$
H	1	2	4	2.2	$[(4 - 1)/6]^2 = 9/36 = 0.25$

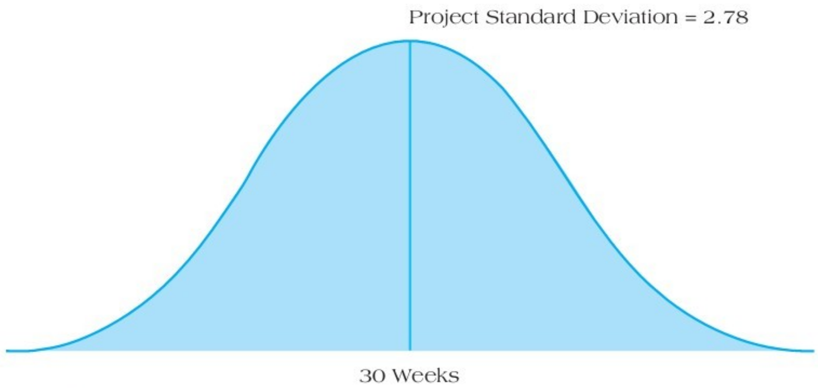


FIGURE 9.21 **Probability Distribution for Project Delta Completion Times**

We can use the information in Table 9.5 to calculate the overall project variance as well. Project variance is found by summing the variances of all *critical* activities and can be represented as the following equation:

$$\sigma_p^2 = \text{Project variance} = \sum(\text{variances of activities on critical path})$$

Thus, using our example, we can calculate the overall project variance and standard deviation for Project Delta. Recall that the critical activities for this project were A – C – D – F – H. For the overall project variance, the calculation is:

$$\text{Project variance } (\sigma_p^2) = 1.78 + 1.00 + 4.00 + .69 + .25 = 7.72$$

The project standard deviation (σ_p) is found as: $\sqrt{\text{Project variance}} = \sqrt{7.72} = 2.78$ weeks.

This project variance information is useful for assessing the probability of on-time project completion because PERT estimates make two more helpful assumptions: (1) Total project completion times use a normal probability distribution, and (2) the activity times are statistically independent. As a result, the normal bell curve shown in Figure 9.21 can be used to represent project completion dates. Normal distribution here implies that there is 50% likelihood that Project Delta's completion time will be less than 30 weeks and a 50% chance that it will be greater than 30 weeks. With this information we are able to determine the probability that our project will be finished on or before a particular time.

Suppose, for example, that it is critical to our company that Project Delta finishes before 32 weeks. Although the schedule calls for a 30-week completion date, remember that our estimates are based on probabilities. Therefore, if we wanted to determine the probability that the project would finish no later than 32 weeks, we would need to determine the appropriate area under the normal curve from Figure 9.22 that corresponds to a completion date on or before week 32. We can use a standard normal equation to determine this probability. The standard normal equation is represented as:

$$\begin{aligned} Z &= (\text{Due date} - \text{Expected date of complection})/\sigma_p \\ &= (32 - 30)/2.78, \text{ or } 0.72 \end{aligned}$$

where Z is the number of standard deviations the target date (32 weeks) lies from the mean or expected date to completion (30 weeks). We can now use a normal distribution table (see Appendix A) to determine that a Z value of 0.72 indicates a probability of 0.7642. Thus, there is a 76.42% chance that Project Delta will finish on or before the critical date of 32 weeks. Visually, this calculation would resemble the picture in Figure 9.22, showing the additional two weeks represented as part of the shaded normal curve to the left of the mean.

Remember from this example that the 32-week deadline is critical for the company to meet. How confident would we be in working on this project if the likelihood of meeting that deadline was only 76.42%? Odds are that the project team (and the organization) might find a 76% chance

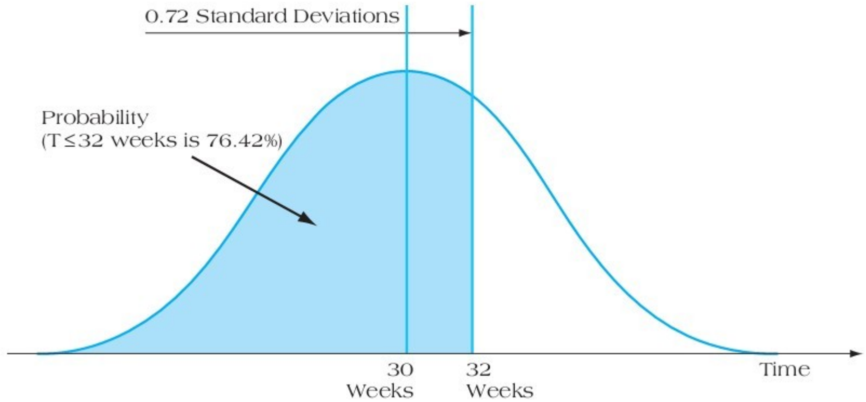


FIGURE 9.22 **Probability of Completing Project Delta by Week 32**

of success in meeting the deadline unacceptable, which naturally leads to the question: How much time will the project team need in order to guarantee delivery with a high degree of confidence?

The first question that needs to be answered is: What is the minimal acceptable likelihood percentage that an organization needs when making this decision? For example, there is a big difference is requiring a 99% confidence of completion versus a 90% likelihood. Let's suppose that the organization developing Project Delta requires a 95% likelihood of on-time delivery. Under this circumstance, how much additional time should the project require to ensure a 95% likelihood of on-time completion?

We are able to determine this value, again, with the aid of Z-score normal distribution tables. The tables indicate that for 95% probability, a Z-score of 1.65 most closely represents this likelihood. We can rewrite the previous standard normal equation and solve for the due date as follows:

$$\begin{aligned} \text{Due Date} &= \text{Expected date of completion} + (Z \times \sigma_p) \\ &= 30 \text{ weeks} + (1.65)\,(2.78) \\ &= 34.59 \text{ weeks} \end{aligned}$$

If the project team can negotiate for an additional 4.59 weeks, they have a very strong (95%) likelihood of ensuring that Project Delta will be completed on time.

It is important to consider one final point regarding estimating probabilities of project completion. So far, we have only addressed activities on the critical path because, logically, they define the overall length of a project. However, there are some circumstances where it may also be necessary to consider noncritical activities and their effect on overall project duration, especially if those activities have little individual slack time and a high variance. For example, in our Project Delta example, activity B has only 1 week of slack and there is sufficient variance of 1.00. In fact, the pessimistic time for activity B is 8 weeks, which would cause the project to miss its target deadline of 30 weeks, even though activity B is not on the critical path. For this reason, it may be necessary to calculate the individual task variances not only for critical activities, but for all project activities, especially those with higher variances. We can then calculate the likelihood of meeting our projected completion dates for all paths, both critical and noncritical.

Laddering Activities

The typical PERT/CPM network operates on the assumption that a preceding activity must be completely finished before the start of the successor task. In many circumstances, however, it may be possible to begin a portion of one activity while work continues on other elements of the task, particularly in lengthy or complex projects. Consider a software development project for a new order-entry system. One task in the overall project network could be to create the Visual Basic code composed of several subroutines to cover the systems of multiple departments. A standard PERT chart would diagram the network logic from coding through debugging as a straightforward logical sequence in which system design precedes coding, which precedes debugging

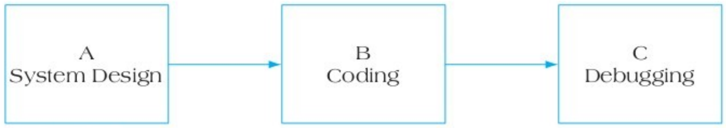


FIGURE 9.23 AON Network for Programming Sequence Without Laddering

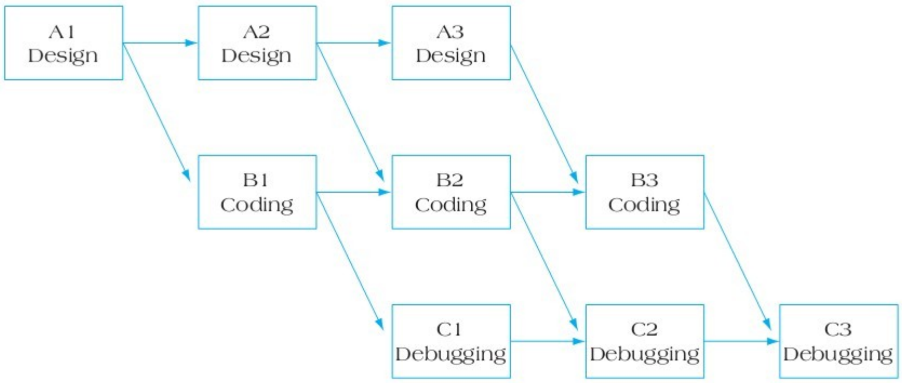


FIGURE 9.24 AON Network with Laddering Effect

(see Figure 9.23). Under severe time pressure to use our resources efficiently, however, we might want to find a method for streamlining, or making the development sequence more efficient.

Laddering is a technique that allows us to redraw the activity network to more closely sequence project subtasks to make the overall network sequence more efficient. Figure 9.24 shows our software development path with laddering. Note that for simplicity's sake, we have divided the steps of design, coding, and debugging into three subtasks. The number of ladders constructed is typically a function of the number of identified break points of logical substeps available. If we assume that the software design and coding project has three significant subroutines, we can create a laddering effect that allows the project team to first complete design phase 1, then move to design phase 2 while coding of design phase 1 has already started. As we move through the laddering process, by the time our designers are ready to initiate design phase 3 in the project, the coders have started on the second subroutine and the debugging staff are ready to begin debugging subroutine 1. The overall effect of laddering activities is to streamline the linkage and sequencing between activities and keep our project resources fully employed.

Hammock Activities

Hammock activities can be used as summaries for some subsets of the activities identified in the overall project network. If the firm needed an outside consultant to handle the coding activities for a software upgrade to its inventory system, a hammock activity within the network can be used to summarize the tasks, duration, and cost. The hammock is so named because it hangs below the network path for consultant tasks and serves as an aggregation of task durations for the activities it "rolls up." Duration for a hammock is found by first identifying all tasks to be included and then subtracting the ES of the first task from the EF of the latest successor. In Figure 9.25, we can see that the hammock's total duration is 26 days, representing a combination of activities D, E, and F with their individual activity durations of 6, 14, and 6 days respectively.

Hammocks allow the project team to better disaggregate the overall project network into logical summaries. This process is particularly helpful when the project network is extremely complex or consists of a large number of individual activities. It is also useful when the project budget is actually shared among a number of cost centers or departments. Hammocking the activities that are assignable to each cost center makes the job of cost accounting for the project much easier.

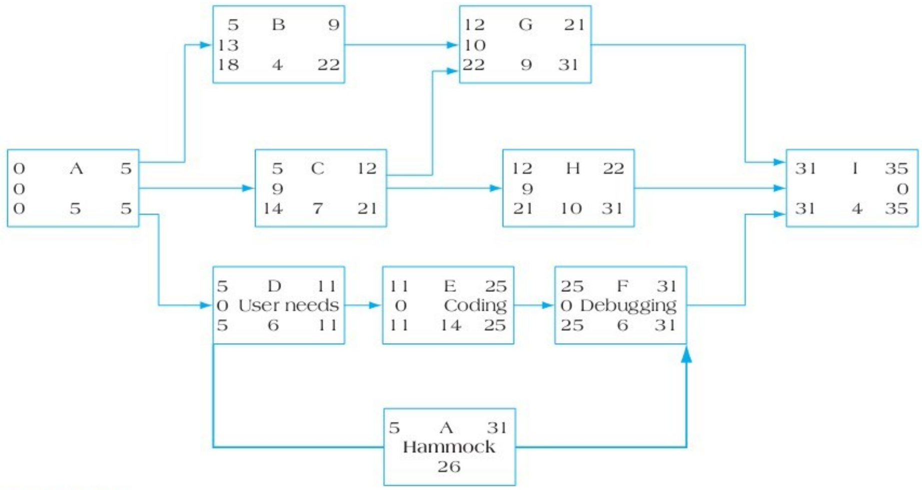


FIGURE 9.25 Example of a Hammock Activity

Options for Reducing the Critical Path

It is common, when constructing an activity network and discovering the expected duration of the project, to look for ways in which the project can be shortened. To do this, start with an open mind to critically evaluate how activity durations were estimated, how the network was originally constructed, and to recognize any assumptions that guided the creation of the network. Reducing the critical path may require several initiatives or steps, but they need to be internally consistent (e.g., their combined effects do not cancel each other out) and logically prioritized.

Table 9.6 shows some of the more common methods for reducing the critical path for a project. The options include not only those aimed at adjusting the overall project network, but also options that address the individual tasks in the network themselves. Among the alternatives for shrinking the critical path are:[9]

1. ***Eliminate tasks on the critical path.*** It may be the case that some of the tasks that are found on the critical path can be eliminated if they are not necessary or can be moved to noncritical paths with extra slack that will accommodate them.
2. ***Replan serial paths to be in parallel.*** In some circumstances, a project may be excessively loaded with serial activities that could just as easily be moved to parallel or concurrent paths in the network. Group brainstorming can help determine alternative methods for pulling serial activities off the critical path and moving them to concurrent, noncritical paths.
3. ***Overlap sequential tasks.*** Laddering is a good method for overlapping sequential activities. Rather than developing a long string of serial tasks, laddering identifies subpoints within the activities where project team members can begin to perform concurrent operations.

TABLE 9.6 Steps to Reduce the Critical Path
1. Eliminate tasks on the critical path.
2. Replan serial paths to be in parallel.
3. Overlap sequential tasks.
4. Shorten the duration of critical path tasks.
5. Shorten early tasks.
6. Shorten longest tasks.
7. Shorten easiest tasks.
8. Shorten tasks that cost the least to speed up.

4. ***Shorten the duration of critical path tasks.*** This option must be explored carefully. The underlying issue here must be to first examine the assumptions that guided the original activity duration estimates for the project. Was beta distribution used reasonably? Were the duration estimates for tasks excessively padded by the project manager or team? Depending upon the answers to these questions, it may indeed be possible to shorten the duration of critical path activities. Sometimes, however, the options of simply shrinking duration estimates by some set amount (e.g., 10% off all duration estimates) all but guarantees that the project will come in behind schedule.
5. ***Shorten early tasks.*** Early tasks in a project are sometimes shortened before later tasks because usually they are more precise than later ones. There is greater uncertainty in a schedule for activities set to occur at some point in the future. Many project managers see that there is likely to be little risk in shortening early tasks, because any lags in the schedule can be made up downstream. Again, however, any time we purposely shorten project activities, we need to be aware of possible ripple effects through the network as these adjustments are felt later.
6. ***Shorten longest tasks.*** The argument for shortening long tasks has to do with relative shrinkage; it is less likely that shortening longer activities will lead to any schedule problems for the overall project network because longer duration tasks can more easily absorb cuts without having an impact on the overall project. For example, shortening a task with 5 days' duration by 1 day represents a 20% cut in the duration estimate. On the other hand, shortening a task of 20 days' duration by 1 day results in only a 5% impact on that activity.
7. ***Shorten easiest tasks.*** The logic here is that the learning curve for a project activity can make it easier to adjust an activity's duration downward. From a cost and budgeting perspective, we saw in Chapter 8 that learning curve methodology does result in lower costs for project activities. Duration estimates for easiest tasks can be overly inflated and can reasonably be lowered without having an adverse impact on the project team's ability to accomplish the task in the shortened time span.
8. ***Shorten tasks that cost the least to speed up.*** "Speeding up" tasks in a project is another way of saying the activities are being crashed. We will cover the process of crashing project activities in more detail in Chapter 10. The option of crashing project activities is one that must be carefully considered against the time/cost trade-off so that the least expensive activities are speeded up.

This chapter has introduced the essential elements in beginning a project schedule, including the logic behind constructing a project network, calculating activity duration estimates, and converting this information into a critical path diagram. These three activities form the core of project scheduling and give us the impetus to begin to consider some of the additional, advanced topics that are important if we are to become expert in the process of project scheduling. These topics will be covered in subsequent chapters.

BOX 9.1

Project Management Research in Brief

Software Development Delays and Solutions

One of the most common problems in IT project management involves the schedule delays found in software development projects. Time and cost overruns in excess of 100% on initial schedules are the industry average. A study by Callahan and Moretton sought to examine how these delays could be reduced. Analyzing the results of 44 companies involved in software development projects, they found that the level of experience firms had with IT project management had a significant impact on the speed with which they brought new products to market. When companies had little experience, the most important action they could take to speed up development times was to interact with customer groups and their own sales organizations early and often throughout the development cycle. The more information they were able to collect on the needs of the customers, the faster they could develop their software products. Also, frequent testing and multiple design iterations were found to speed up the delivery time.

For firms with strong experience in developing software projects, the most important determinants of shorter development cycles were found to be developing relationships with external suppliers, particularly during the product requirements, system design, and beta testing phases of the project. Supplier involvement in all phases of the development cycle proved to be key to maintaining aggressive development schedules.[10]

Summary

1. **Understand and apply key scheduling terminology.** Key processes in project scheduling include how activity networks are constructed, task durations are estimated, the critical path and activity float are calculated, and lag relationships are built into activities.
2. **Apply the logic used to create activity networks, including predecessor and successor tasks.** The chapter discussed the manner in which network logic is employed. Following the creation of project tasks, through use of Work Breakdown Structures, it is necessary to apply logic to these tasks in order to identify those activities that are considered predecessors (coming earlier in the network) and those that are successors (coming later, or after the predecessor activities have been completed).
3. **Develop an activity network using Activity-on-Node (AON) techniques.** The chapter examined the process for creating an AON network through identification of predecessor relationships among project activities. Once these relationships are known, it is possible to begin linking the activities together to create the project network. Activity-on-Node (AON) applies the logic of assigning all tasks as specific "nodes" in the network and linking them with arrows to identify the predecessor-successor relationships.
4. **Perform activity duration estimation based on the use of probabilistic estimating techniques.** Activity duration estimation is accomplished through first identifying the various tasks in a project and then applying a method for estimating the duration of each of these activities. Among the methods that can aid us in estimating activity durations are (1) noncomputational techniques, for example, examining past records for similar tasks that were performed at other times in the organization and obtaining expert opinion; (2) deriving duration estimates through computational, or mathematical, analysis; and (3) using the Program Evaluation and Review Technique (PERT), which uses probabilities to estimate a task's duration. In applying PERT, the formula for employing a beta probability distribution is to first determine optimistic, most likely, and pessimistic estimates for the duration of each activity and then assign them in a ratio of:

$$[(1 \times \text{optimistic}) + (4 \times \text{most likely}) + (1 \times \text{pessimistic})]/6$$

5. **Construct the critical path for a project schedule network using forward and backward passes.** Conducting the forward pass allows us to determine the overall expected duration for the project by using the decision rules, adding early start plus activity duration to determine early finish, and then applying this early finish value to the next node in the network, where it becomes that activity's early start. We then use our decision rules for the backward pass to identify all activities and paths with float and the project's critical path (the project path with no float time).
6. **Identify activity float and the manner in which it is determined.** Once the network linking all project activities has been constructed, it is possible to begin determining the estimated duration of each activity. Duration estimation is most often performed using probabilistic estimates based on Program Evaluation and Review Technique (PERT) processes, in which optimistic, most likely, and pessimistic duration estimates for each activity are collected. Using a standard formula based on the statistically derived beta distribution, project activity durations for each task are determined and used to label the activity nodes in the network.

   Using activity durations and the network, we can identify the individual paths through the network, their lengths, and the critical path. The project's critical path is defined as the activities of a project which, when linked, define its shortest total length. The critical path identifies how quickly we can complete the project. All other paths contain activities that have, to some degree, float or slack time associated with them. The identification of the critical path and activity float times is done through using a forward and backward pass process in which each activity's early start (ES), early finish (EF), late start (LS), and late finish (LF) times are calculated.
7. **Calculate the probability of a project finishing on time under PERT estimates.** Because PERT estimates are based on a range of estimated times (optimistic, most likely, pessimistic), there will be some variance associated with these values and expected task duration. Determining the variance of all activities on the critical (and noncritical) paths allows us to more accurately forecast the probability of completing the project on or before the expected finish date. We can also use the standard normal equation (and associated Z score) to forecast the additional time needed to complete a project under different levels of overall confidence.
8. **Understand the steps that can be employed to reduce the critical path.** Project duration can be reduced through a number of different means. Among the options project managers have to shorten the project critical path are the following: (1) Eliminate tasks on the critical path, (2) replan serial paths to be in parallel, (3) overlap sequential tasks, (4) shorten the duration of critical path tasks, (5) shorten early tasks, (6) shorten longest tasks, (7) shorten easiest tasks, and (8) shorten tasks that cost the least to speed up. The efficacy of applying one of these approaches over another will vary depending on a number of issues related both to the project constraints, client expectations, and the project manager's own organization.

Key Terms

Solved Problems

9.1 CREATING AN ACTIVITY NETWORK

Assume the following information:

Activity	Predecessors
A	—
B	A
C	B
D	B
E	C, D
F	C
G	E, F
H	D, G

Create an activity network that shows the sequential logic between the project tasks. Can you identify merge activities? Burst activities?

SOLUTION

This activity network can be solved as shown in Figure 9.26. The merge points in the network are activities E, G, and H. The burst activities are activities B, C, and D.

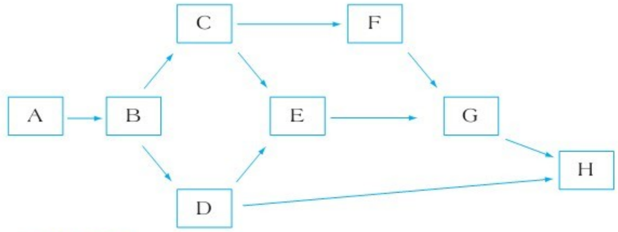


FIGURE 9.26 Solution to Solved Problem

9.2 CALCULATING ACTIVITY DURATIONS AND VARIANCES

Assume that you have the following pessimistic, likely, and optimistic estimates for how long activities are estimated to take. Using the beta distribution, estimate the activity durations and variances for each task.

	Duration Estimates		
Activity	**Pessimistic**	**Likely**	**Optimistic**
A	7	5	2
B	5	3	2
C	14	8	6
D	20	10	6
E	8	3	3
F	10	5	3
G	12	6	4
H	16	6	5

SOLUTION

Remember that the beta distribution calculates expected activity duration (TE) as:

$$\text{TE} = (a + 4m + b)/6$$

Where
TE = estimated time for activity
a = most optimistic time to complete the activity
m = most likely time to complete the activity, the mode of the distribution
b = most pessimistic time to complete the activity

The formula for activity variance is:

$$s^2 = [(b - a)/6]^2$$

Therefore, in calculating expected activity duration (TE) and variance for each task we find the value as shown in the table below.

	Duration Estimates				
Activity	Pessimistic	Likely	Optimistic	TE (Beta)	Variance
A	7	5	2	4.8	$[(7-2)/6]^2 = 25/36 = 0.69$
B	5	3	2	3.2	$[(5-2)/6]^2 = 9/36 = 0.25$
C	14	8	6	8.7	$[(14-6)/6]^2 = 64/36 = 1.78$
D	20	10	6	11.0	$[(20-6)/6]^2 = 196/36 = 5.44$
E	8	3	3	3.8	$[(8-3)/6]^2 = 25/36 = 0.69$
F	10	5	3	5.5	$[(10-3)/6]^2 = 49/36 = 1.36$
G	12	6	4	6.7	$[(12-4)/6]^2 = 64/36 = 1.78$
H	16	6	5	7.5	$[(16-5)/6]^2 = 121/36 = 3.36$

9.3 DETERMINING CRITICAL PATH AND ACTIVITY SLACK

Assume we have a set of activities, their expected durations, and immediate predecessors. Construct an activity network; identify the critical path and all activity slack times.

Activity	Predecessors	Expected Duration
A	—	6
B	A	7
C	A	5
D	B	3
E	C	4
F	C	5
G	D, E	8
H	F, G	3

SOLUTION

We follow an iterative process of creating the network and labeling the nodes as completely as possible. Then, following Figure 9.27, we first conduct a forward pass through the network to determine that the expected duration of the project is 27 days. Using a backward pass, we can determine the individual activity slack times as well as the critical path. The critical path for this example is as follows: A – B – D – G – H. Activity slack times are:

C = 1 day
E = 1 day
F = 8 days

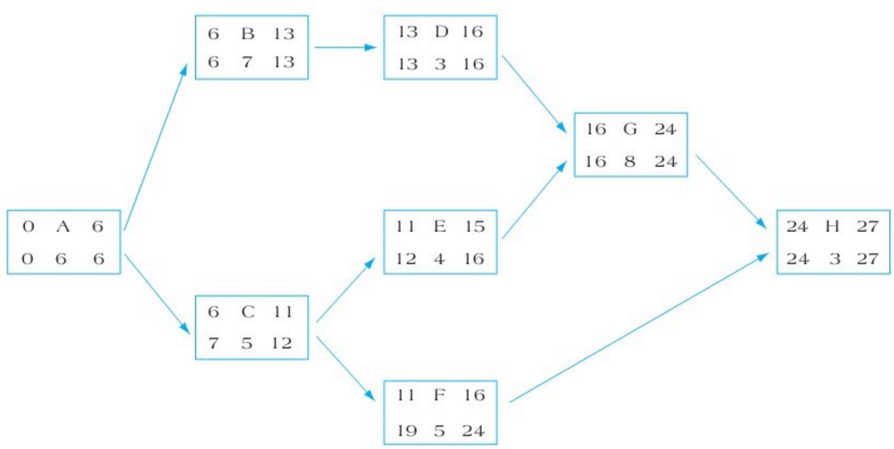


FIGURE 9.27 Solution to Solved Problem 9.3