sis phase is described in more detail. For example, data gathering is broken down into five activities, from conducting interviews to observing reactions to the prototype. This particular project requires data flow analysis but not decision analysis, so the systems analyst has written in "analyze data flow" as the single step in the middle phase. Finally, proposal preparation is broken down into three steps: perform cost-benefit analysis, prepare proposal, and present proposal.

The systems analyst, of course, has the option to break down steps further. For instance, the analyst could specify each of the persons to be interviewed. The amount of detail necessary depends on the project, but all critical steps need to appear in the plans.

Sometimes the most difficult part of project planning is the crucial step of estimating the time it takes to complete each task or activity. When quizzed about reasons for lateness on a particular project, project team members cited poor scheduling estimates that hampered the success of projects from the outset. There is no substitute for experience in estimating time requirements, and systems analysts who have had the opportunity of an apprenticeship are fortunate in this regard.

Planners have attempted to reduce the inherent uncertainty in determining time estimates by projecting most likely, pessimistic, and optimistic estimates and then using a weighted average formula to determine the expected time an activity will take. This approach offers little more in the way of confidence, however. Perhaps the best strategy for the systems analyst is to adhere to a structured approach in identifying activities and describing these activities in sufficient detail. In this manner, the systems analyst will at least be able to limit unpleasant surprises.

Using Gantt Charts for Project Scheduling

A Gantt chart is an easy way to schedule tasks. It is a chart on which bars represent each task or activity. The length of each bar represents the relative length of the task.

Figure 3.18 is an example of a two-dimensional Gantt chart in which time is indicated on the horizontal dimension and a description of activities makes up the vertical dimension. In this example the Gantt chart shows the analysis or information gathering phase of the project. Notice on the Gantt chart that conducting interviews will take three weeks, administering the questionnaire will take four weeks, and so on. These activities overlap part of the time. In the chart the special symbol ▲ signifies that it is week 9. The bars with color shading represent projects or parts of projects that have been completed, telling us that the systems analyst is behind in introducing prototypes but ahead in analyzing data flows. Action must be taken on introducing prototypes soon so that other activities or even the project itself will not be delayed as a result.

The main advantage of the Gantt chart is its simplicity. The systems analyst will find not only that this technique is easy to use but also that it lends itself to worthwhile communication with end users. Another advantage of using a Gantt chart is that the bars representing activities or tasks are drawn to scale; that is, the size of the bar indicates the relative length of time it will take to complete each task.

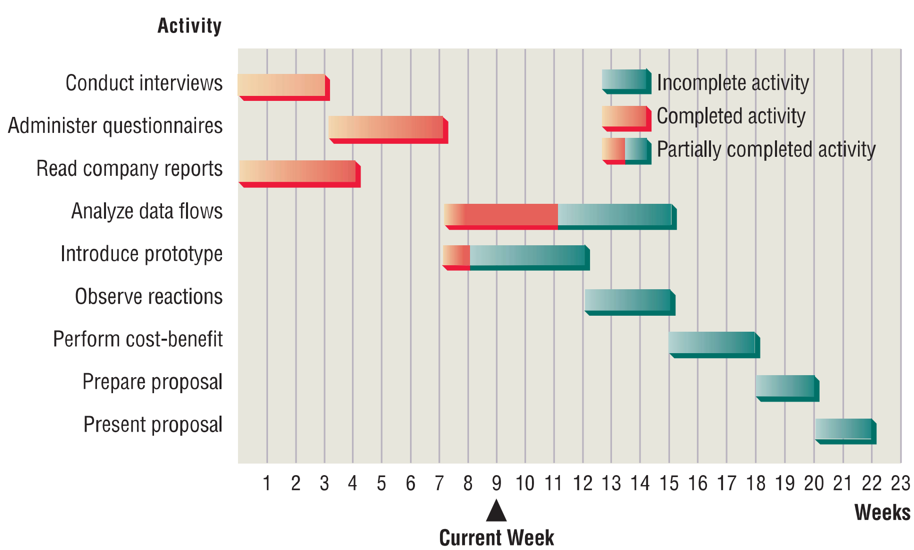

FIGURE 3.18

Using a two-dimensional Gantt chart for planning activities that can be accomplished in parallel.

Using PERT Diagrams

PERT is an acronym for Program Evaluation and Review Techniques. A program (a synonym for a project) is represented by a network of nodes and arrows that are then evaluated to determine the critical activities, improve the schedule if necessary, and review progress once the project is undertaken. PERT was developed in the late 1950s for use in the U.S. Navy's Polaris nuclear submarine project. It reportedly saved the U.S. Navy two years' development time.

PERT is useful when activities can be done in parallel rather than in sequence. The systems analyst can benefit from PERT by applying it to systems projects on a smaller scale, especially when some team members can be working on certain activities at the same time that fellow members are working on other tasks.

Figure 3.19 compares a simple Gantt chart with a PERT diagram. The activities expressed as bars in the Gantt chart are represented by arrows in the PERT diagram. The length of the arrows has no direct relationship with the activity durations. Circles on the PERT diagram are called events and can be identified by numbers, letters, or any other arbitrary form of designation. The circular nodes are present to (1) recognize that an activity is completed and (2) indicate which activities need to be completed before a new activity may be undertaken (precedence).

In reality activity C may not be started until activity A is completed. Precedence is not indicated at all in the Gantt chart, so it is not possible to tell whether activity C is scheduled to start on day 4 on purpose or by coincidence.

A project has a beginning, a middle, and an end; the beginning is event 10 and the end is event 50. To find the length of the project, each path from beginning to end is identified, and the length of each path is calculated. In this example path 10–20–40–50 has a length of 15 days, whereas path 10–30–40–50 has a length of 11 days. Even though one person may be working on path 10–20–40–50 and another on path 10–30–40–50, the project is not a race. The project requires that both sets of activities (or paths) be completed; consequently, the project takes 15 days to complete.

The longest path is referred to as the critical path. Although the critical path is determined by calculating the longest path, it is defined as the path that will cause the whole project to fall behind if even one day's delay is encountered on it. Note that if you are delayed one day on path 10–20–40–50, the entire project will take longer, but if you are delayed one day on path 10–30–40–50, the entire project will not suffer. The leeway to fall behind somewhat on noncritical paths is called slack time.

Occasionally, PERT diagrams need pseudo-activities, referred to as dummy activities, to preserve the logic of or clarify the diagram. Figure 3.20 shows two PERT diagrams with dummies. Project 1 and project 2 are quite different, and the way the dummy is drawn makes the difference

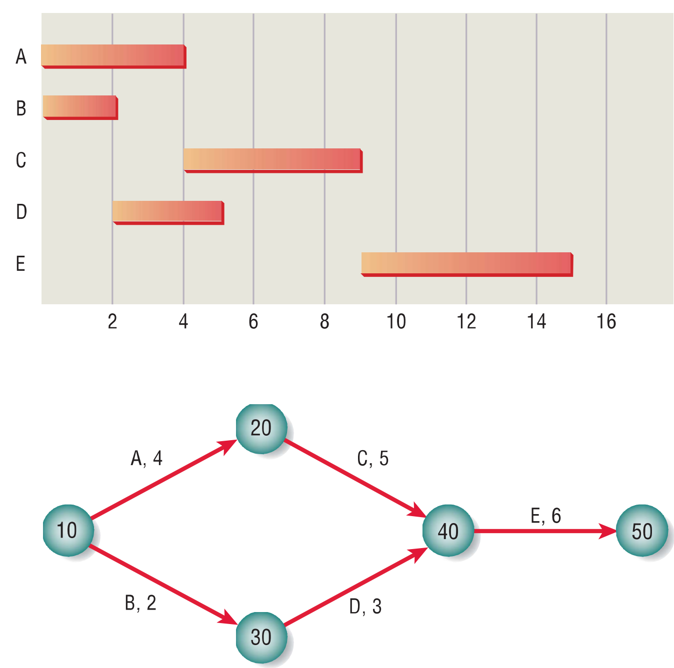

FIGURE 3.19

Gantt charts compared with PERT diagrams for scheduling activities.

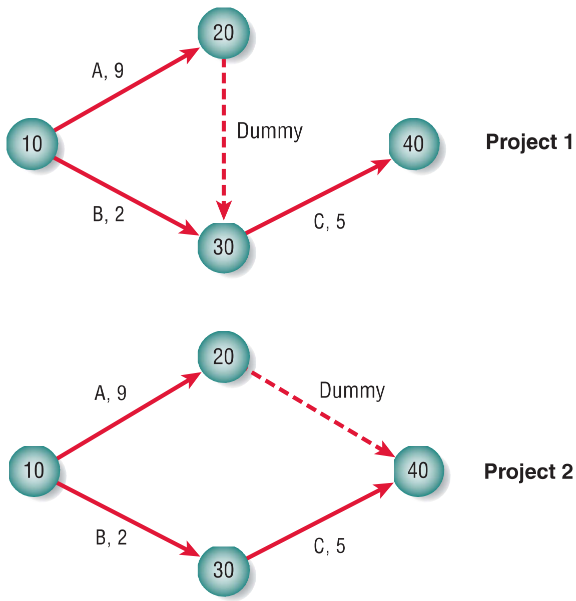

FIGURE 3.20

Precedence of activities is important in determining the length of the project when using a PERT diagram.

clear. In project 1 activity C can only be started if both A and B are finished, because all arrows coming into a node must be completed before leaving the node. In project 2, however, activity C requires only activity B's completion and can therefore be under way while activity A is still taking place.

Project 1 takes 14 days to complete, whereas project 2 takes only 9 days. The dummy in project 1 is necessary, of course, because it indicates a crucial precedence relationship. The dummy in project 2, on the other hand, is not required, and activity A could have been drawn from 10 to 40 and event 20 may be eliminated completely.

Therefore, there are many reasons for using a PERT diagram over a Gantt chart. The PERT diagram allows:

1. Easy identification of the order of precedence.
2. Easy identification of the critical path and thus critical activities.
3. Easy determination of slack time.

A PERT EXAMPLE. Suppose a systems analyst is trying to set up a realistic schedule for the data gathering and proposal phases of the systems analysis and design life cycle. The systems analyst looks over the situation and lists activities that need to be accomplished along the way. This list, which appears in Figure 3.21, also shows that some activities must precede other activities. The time estimates were determined as discussed in an earlier section of this chapter.

DRAWING THE PERT DIAGRAM. In constructing the PERT diagram, the analyst looks first at those activities requiring no predecessor activities, in this case A (conduct interviews) and C (read company reports). In the example in Figure 3.22, the analyst chose to number the nodes 10, 20, 30, and so on, and he or she drew two arrows out of the beginning node 10. These arrows represent

Activity		Predecessor	Duration
A	Conduct interviews	None	3
B	Administer questionnaires	A	4
C	Read company reports	None	4
D	Analyze data flow	B, C	8
E	Introduce prototype	B, C	5
F	Observe reactions to prototype	E	3
G	Perform cost-benefit analysis	D	3
H	Prepare proposal	F, G	2
I	Present proposal	H	2

FIGURE 3.21

Listing activities for use in drawing a PERT diagram.

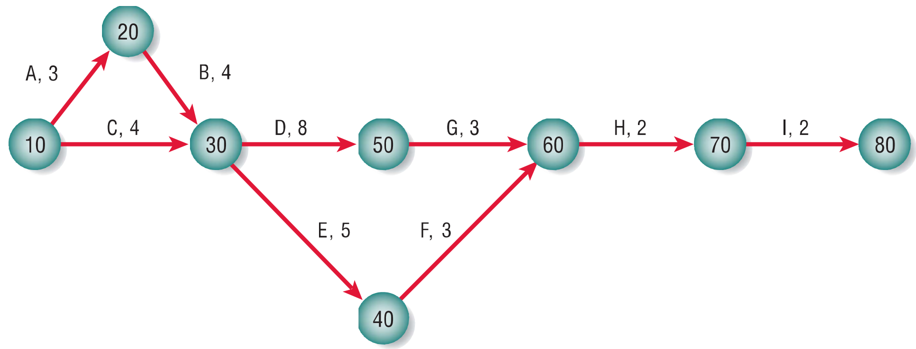

FIGURE 3.22

A completed PERT diagram for the analysis phase of a systems project.

activities A and C and are labeled as such. Nodes numbered 20 and 30 are drawn at the end of these respective arrows. The next step is to look for any activity requiring only A as a predecessor; task B (administer questionnaires) is the only one, so it can be represented by an arrow drawn from node 20 to node 30.

Because activities D (analyze data flow) and E (introduce prototype) require both activities B and C to be finished before they are started, arrows labeled D and E are drawn from node 30, the event that recognizes the completion of both B and C. This process is continued until the entire PERT diagram is completed. Notice that the entire project ends at an event called node 80.

IDENTIFYING THE CRITICAL PATH. Once the PERT diagram is drawn, it is possible to identify the critical path by calculating the sum of the activity times on each path and choosing the longest path. In this example, there are four paths: 10–20–30–50–60–70–80, 10–20–30–40–60–70–80, 10–30–50–60–70–80, and 10–30–40–60–70–80. The longest path is 10–20–30–50–60–70–80, which takes 22 days. It is essential that the systems analyst carefully monitors the activities on the critical path so as to keep the entire project on time or even shorten the project length if warranted.

MANAGING THE PROJECT

The process of analysis and design can become unwieldy, especially when the system being developed is large. To keep the development activities as manageable as possible, we usually employ some of the techniques of project management to help us get organized.

One important aspect of project management is how to manage one's schedule to finish the system on time, but it is not the only thing needed. The person in charge, called the project manager, is often the lead systems analyst. The project manager needs to understand how to determine what is needed and how to initiate a project; how to develop a problem definition; how to examine feasibility of completing the systems project; how to reduce risk; how to identify and manage activities; and how to hire, manage, and motivate other team members.

Addressing System Complexity

Estimating models, such as Costar (www.softstarsystems.com) or Construx (www.construx.com), work as follows: First the systems analyst enters an estimate of the size of the system. This can be entered in a number of different ways, including the lines of source code of the current system. Then it may be helpful to adjust the degree of difficulty based on how familiar the analyst is with this type of project.

Also considered are other variables, like the experience or capability of the team, the type of platform or operating system, the level of usability of the finished software (for example, what languages are necessary), and other factors that can drive up costs. Once the data are entered, calculations are made, and a rough projection of the completion date is produced. As the project gets underway, more specific estimates are possible.

Another way of estimating the amount of work that needs to be done and how large a staff one needs to complete a project is called function point analysis. This method takes the five main components of a computer system—(1) external inputs, (2) external outputs, (3) external queries, (4) internal logical files, and (5) external interface files— and then rates them in terms of complexity.

Function point analysis can estimate the time it takes to develop a system in different computer languages and compare them to one another. For more information about function point analysis, visit the International Function Point Users Group's Web site at www.ifpug.org.