

PROJECT SCHEDULES WITH PERT/CPM CHARTS

Chapter 3 of *Systems Analysis and Design in a Changing World* explains the techniques and steps required to build a project schedule using the Gantt chart view in MS Project. This appendix provides a similar explanation of how to build a project schedule, but it is based on using a PERT/CPM chart or diagram for the schedule format. A Gantt chart and a PERT/CPM chart both provide essentially the same information about project activities and tasks. Each chart has unique strengths and weaknesses.

As you learned in Chapter 3, a Gantt chart is a type of bar chart that is superimposed on a calendar. The primary strength of a Gantt chart is that the bars show the duration and project progress as compared to the days and weeks of the calendar. The calendar comparison provides an effective visual representation of the project timeline and helps you track project progress, because the Gantt chart shows what should have been accomplished by a specific date and what has actually been completed on that date. Chapter 3 illustrates an example of the tracking view of a Gantt chart.

A PERT/CPM chart, as you will learn in this appendix, is a network type of diagram with boxes that represent the tasks or activities of the project, and with connecting arrows that represent the sequence and dependencies between tasks. The strength of a PERT/CPM chart is that, as a network, it provides a visual representation of the relationships between tasks of a project. During the development of the project schedule, and especially while trying to determine task dependencies, a PERT/CPM chart is an effective tool.

PERT, which stands for Project Evaluation and Review Technique, was first developed in the 1950s and was used by the United States Department of Defense to organize, monitor, and control very large, complex defense projects. CPM, which stands for Critical Path Method, was developed independently, also in the 1950s. As its name implies, its primary objective was to determine not only the dependencies between tasks, but which tasks were on the critical path. These two methods have much in common, and in recent years most scheduling tools have taken the best attributes of each one and combined them into a single technique—thus the name PERT/CPM.

In this appendix, we will first develop a PERT/CPM chart by hand so that you can learn and understand the basic concepts. Then we will illustrate a PERT/CPM chart in MS Project.

Developing a PERT/CPM chart is a five-step process:

1. Identify all the tasks for the project (that is, build a work breakdown structure or WBS).
2. Determine the amount of work necessary to complete each task.
3. For each task, identify the immediate predecessor tasks.
4. Enter the tasks on a PERT/CPM chart, with connecting arrows for task dependencies.
5. Calculate start and end times based on durations and resources.

As discussed in Chapter 3, the first three steps are always manual tasks and are often done in a brainstorming session. The last two steps happen automatically when entering the information into a scheduling tool such as MS Project. In Chapter 3, we provided guidelines in identifying and delimiting tasks. The chapter explained how to enter data for Steps 2 and 3, but detailed explanations were not included. We provide some additional guidance for those steps in this appendix.

WORK BREAKDOWN STRUCTURE

Let's start with the same set of tasks we identified in Chapter 3. Figure B-1 is a copy of that information in a document format. The first step—identifying all the tasks—can be done using any of the approaches explained in Chapter 3. Note that this example uses a hierarchical structure and numbers the tasks to show this hierarchy. By grouping individual tasks together into a larger activity, project managers can define summary activities and larger milestones that help to monitor the progress of the project.

Figure B-1

Work breakdown structure for the project planning activities for RMO

1. Project Planning
1.1 Define the problem
1.1.1 Meet with users
1.1.2 Determine scope
1.1.3 Write problem description
1.1.4 Identify business benefits
1.1.5 Identify system capabilities
1.1.6 Develop context diagram
1.2 Produce project schedule
1.2.1 Develop WBS
1.2.2 Estimate durations
1.2.3 Determine sequences
1.2.4 Develop PERT/CPM chart
1.3 Confirm project feasibility
1.3.1 Identify intangible cost/benefits
1.3.2 Estimate tangible benefits
1.3.3 Calculate cost/benefit
1.3.4 Organizational feasibility
1.3.5 Technical feasibility
1.3.6 Evaluate resource availability
1.3.7 Risk analysis
1.4 Staff the project
1.4.1 Develop resource plan
1.4.2 Procure project team
1.4.3 Procure user liaisons
1.4.4 Conduct training
1.5 Launch the project
1.5.1 Make executive presentation
1.5.2 Procure facilities
1.5.3 Procure support resources
1.5.4 Conduct kickoff meeting



As mentioned earlier, PERT and CPM have essentially merged into a single scheduling technique. One basic difference between them was in the techniques used to estimate the effort required for the tasks. CPM used a simple estimate of the effort required, while PERT used the most probable effort for the task. With PERT, three estimates are made for the task: pessimistic, most likely, and optimistic. The three are combined, using a weighted averaging scheme, into a single value to give the expected effort required for the task. A weighted average simply ensures that any large deviation by either the optimistic or pessimistic estimates will not drastically move the most probable duration away from its expected duration. However, this technique is not used very often today, especially for software projects. Today, most managers simply make their best estimate for each task.

We emphasize, however, that project managers should not make these estimates in a vacuum. In fact, if other people are already on the team, the preferable method is to involve team members who will be assigned to the task and have them assist in estimating the effort. More realistic estimates are obtained by having the right people involved. In Figure B-2, we update the individual tasks with estimated durations and resources requirements. The first task requires two days and the second one day, each with two people working. Thus, the first task requires four person-days of work and the second requires two. No duration is assigned to the summary categories because their duration is the composite of the individual tasks.

Figure B-2

Work breakdown structure with duration and effort estimates

1. Project Planning	Duration in days	Resources required
1.1 Define the problem		
1.1.1 Meet with users	2	2
1.1.2 Determine scope	1	2
1.1.3 Write problem description	1	1
1.1.4 Identify business benefits	2	1
1.1.5 Identify system capabilities	1	1
1.1.6 Develop context diagram	2	1
1.2 Produce project schedule		
1.2.1 Develop WBS	3	2
1.2.2 Estimate durations	1	2
1.2.3 Determine sequences	2	2
1.2.4 Develop PERT/CPM chart	3	1
1.3 Confirm project feasibility		
1.3.1 Identify intangible cost/benefits	1	2
1.3.2 Estimate tangible benefits	1	2
1.3.3 Calculate cost/benefit	1	2
1.3.4 Organizational feasibility	2	2
1.3.5 Technical feasibility	2	2
1.3.6 Evaluate resource availability	2	2
1.3.7 Risk analysis	1	2
1.4 Staff the project		
1.4.1 Develop resource plan	2	2
1.4.2 Procure project team	2	1
1.4.3 Procure user liaisons	3	1
1.4.4 Conduct training	4	2
1.5 Launch the project		
1.5.1 Make executive presentation	3	2
1.5.2 Procure facilities	2	1
1.5.3 Procure support resources	2	2
1.5.4 Conduct kickoff meeting	1	2

PERT/CPM CHART

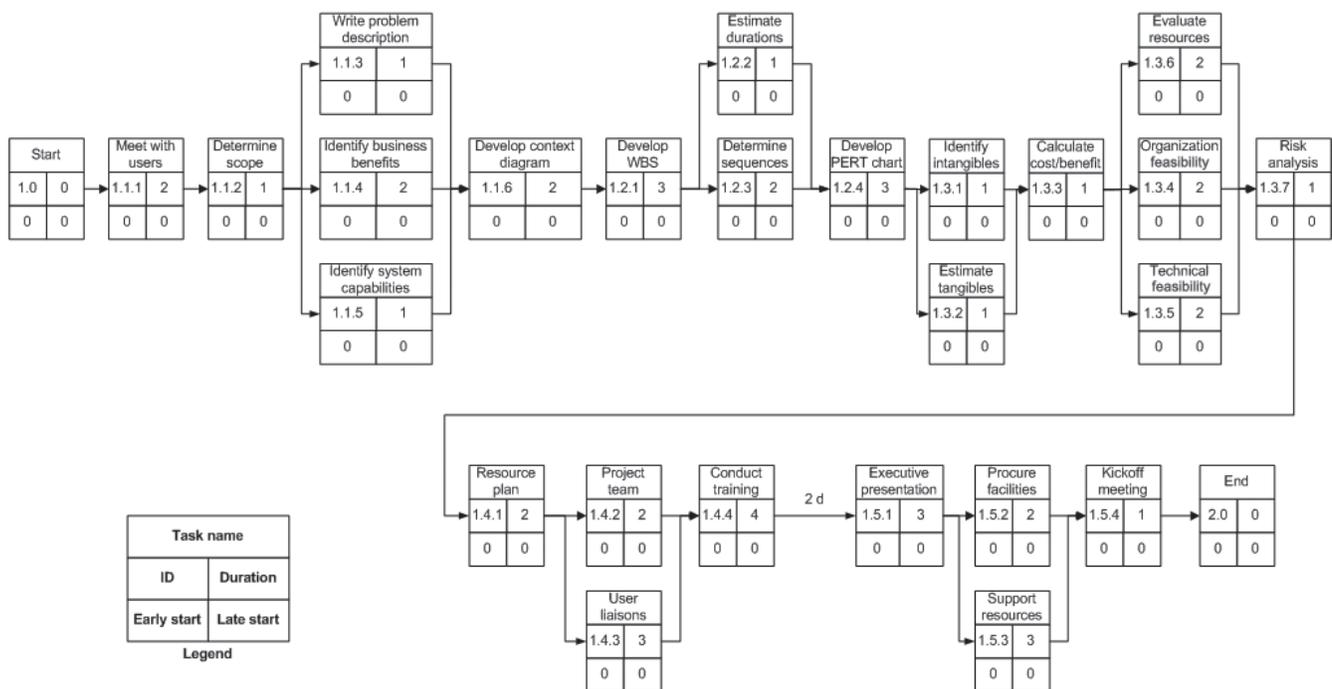
The third step, assigning task dependencies, lays the foundation for the development of the PERT /CPM chart. There are three types of dependencies between tasks: (1) project mandatory,

(2) external mandatory, and (3) discretionary. Predecessors usually are easily assigned to project mandatory dependencies. One task depends on the completion or output of another task and thus is dependent on it. For example, an acceptance test cannot begin until the acceptance test criteria and data have been defined (at least partially). External mandatory dependencies are caused by other, nonproject occurrences, such as the arrival of new hardware or the completion of an interface definition from another project. The most troublesome dependencies are the discretionary ones. They provide the most flexibility for building the schedule but they also add the most complexity. For example, developers must determine which should be done first, designing input screens or output reports. Either will probably work. Complexity is added when the project manager tries to balance the scheduling of these tasks with the available resources (that is, the team members). Often, several tentative schedules must be built with different dependencies to determine a sequence that best uses team member skills. This activity is often best completed with a whiteboard and Post-it notes. The individual tasks can be placed on the board with positions based on an approximate dependency. Arrows can then be drawn to identify the best combination of dependencies to balance workloads.

For now, let's assume that all the juggling and rearranging has been done, and that the information from the whiteboard has been captured in a computer format. If you are using MS Project, you would enter this information into the program, as explained in Chapter 3. However, let's first do a manual example to better understand the underlying concepts. Figure B-3 is a sample PERT/CPM chart that could have been drawn with any computer drawing tool. In this instance, we use rectangles with five compartments to document each task. To do this with a drawing tool, we first create a prototype rectangle and then cut and paste it multiple times. The legend shows the meaning of each compartment.

Figure B-3

First-cut manual PERT/CPM chart for RMO planning activities



The tasks are represented by the rectangles and the predecessor/successor relationships are indicated by the connecting arrows. Again, this information is captured directly from a whiteboard working session. In this example, we have used the same hierarchical numbering scheme from the WBS for the task ID. The summary activities have not been included because they do not participate directly in the dependency list. However, start and end “dummy” tasks have been added for ease of understanding. The compartments within the rectangles display information about the tasks.

The format shown in Figure B-3 is just one of many possible formats. The MS Project format shown later is slightly different. The top compartment within the rectangle is the task name. The left-middle compartment is the task ID, and the right-middle compartment is the duration. The bottom-left corner is the early start time for each task, and the bottom-right corner is the late start for each task. To keep this example simple, the duration numbers represent days, and the start time for each task is the number of days from the beginning of the project. Optionally, the early start and late start can also be shown as actual calendar dates with an automated tool such as MS Project. Next, we explain and calculate the early and late start times.

early start time

the earliest time that a task can begin based on completion times of predecessor tasks

CALCULATING EARLY START TIMES

The bottom two compartments in Figure B-3 are empty. The next step in our manual process is to calculate the numbers for those compartments. The bottom-left compartment contains a number that is called the **early start time** for the task. The early start time represents the earliest time that a task can begin. In other words, based on all of the predecessor tasks, the given task cannot begin until the time indicated. This field is calculated by taking the early start time of the predecessor task and adding to it the duration of the predecessor task. Here is the formula:

$$ES_{n+1} = ES_n + Duration_n + lag\ time$$

This formula indicates that the early start time of a successor task (for example, $n+1$) is equal to the early start of the n^{th} task plus the duration of the n^{th} task, plus any lag time if it is specified. Figure B-4 contains the PERT/CPM chart with all of the early start times calculated. To calculate early start times, we begin with the first detailed task, number 1.1.1. In this case, $ES_n = 0$, $Duration_n = 2$, and consequently $ES_{n+1} = 2$. The rest of the early start times are calculated one at a time, moving from beginning to end and following the arrows through the chart. Lag time is associated with the relationship (the arrow) between the two tasks. An example of lag time is shown between tasks 1.4.4 and 1.5.1.

Recall that in Chapter 3 we identified several types of predecessor/successor relationships: finish-start, start-start, and finish-finish relationships. The preceding equation works only for the finish-start relationship because the successor task starts after the predecessor task is finished. Equations for other relationships are:

$$ES_{n+1} = ES_n + lag\ time\ (for\ start-start),\ and$$

$$ES_{n+1} + Duration_{n+1} = ES_n + Duration_n + lag\ time\ (for\ finish-finish).$$

All of these times are calculated with a forward pass through the chart, starting at the first task and moving sequentially following the arrows (that is, left to right) through the tasks.

In this example, the tasks numbered 1.1.3 through 1.1.5 can all occur concurrently, which illustrates some of the complexity of identifying dependencies among the tasks. As the project manager was building the dependencies, she noticed that these three tasks are interrelated and should be worked on at the same time. Each of these tasks will have the same early start time because they all follow the same predecessor task of 1.1.2.

The next complexity to address is how to handle successor tasks of a set of concurrent tasks, in this case task 1.1.6. Obviously, task number 1.1.6 cannot begin until the latest predecessor task finishes. Let's modify the preceding equation to be the following:

$$ES_{n+1} = \{ES_n + Duration_n\}_{latest} + lag\ time$$

In this equation, the curly braces indicate a set of values, or set of concurrent tasks, and the subscript "latest" indicates that we choose the latest one as the correct value for the equation.

This project requires 38 days to complete, which is the value in the final "End" task. Be sure you understand how all the values were calculated, including both examples with concurrent tasks and the one example with a lag time.

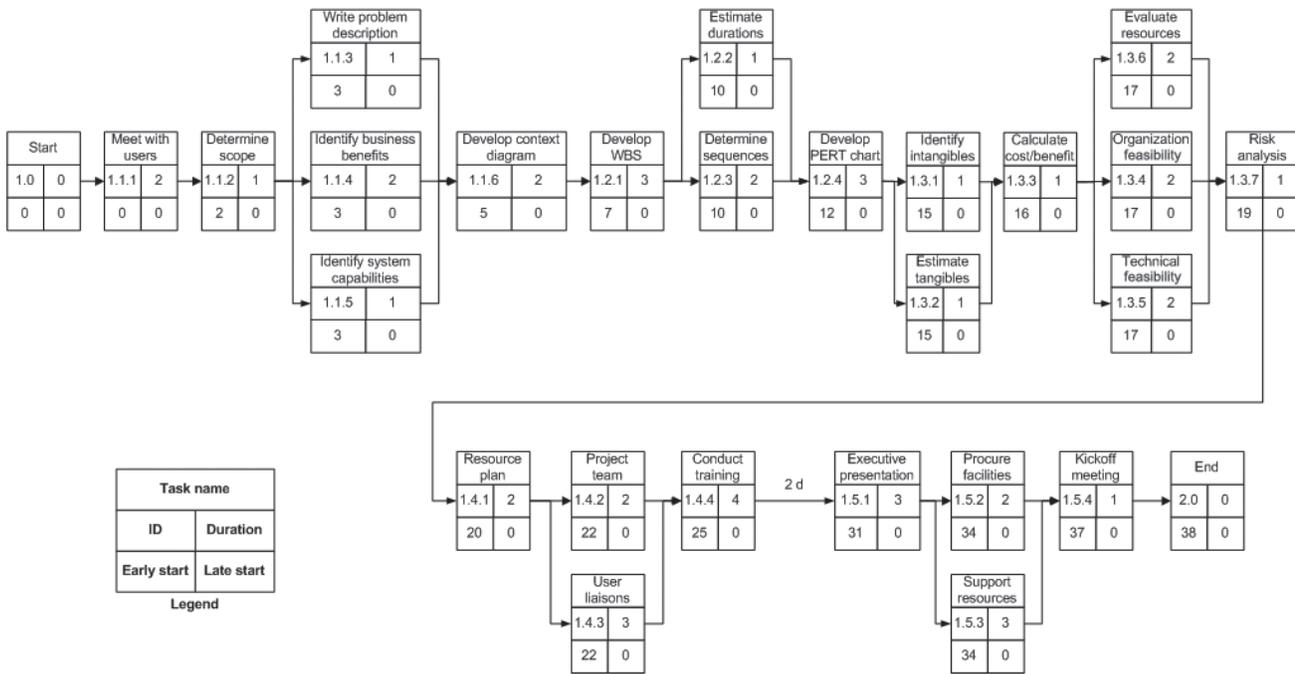


Figure B-4

PERT/CPM chart for RMO with early start times calculated

late start time

the latest time that a task can begin and still keep the project on schedule

CALCULATING LATE START TIMES

The bottom-right compartments contain the times for the **late start time**. Not all formats of PERT/CPM diagrams show the late start time, but this concept will help us understand some other more advanced concepts. The late start time is the latest time that a given task can start and *still keep the project on schedule*. Late start times are calculated starting at the end of the project and moving forward. We start by noting that the final task, or end task, has an early start (and finish because it has zero duration) on day 38. Move that same value into the lower-right compartment, which denotes that the late start time is the same as the early start time. To calculate other late start times, we simply move backward through the diagram using the following equation:

$$LS_{n-1} + duration_{n-1} = LS_n \quad \text{or equivalently} \quad LS_{n-1} = LS_n - duration_{n-1}$$

This equation shows that the late start of a predecessor task is equal to the late start of a successor task, minus the duration of the predecessor task. It is similar to the early start calculation, except that we are moving backward through the diagram.

Two special situations must be handled, just as in our previous examples: concurrent tasks and lag times. Let's make the preceding equation more general.

$$LS_{n-1} = \{LS_n - duration_{n-1}\}_{earliest} - lag\ time$$

As before, we calculate the late start for each leg, and then choose the earliest time (the smallest number) as the correct one. The lag time associated with the arrow is also subtracted, because it is time that must be removed from the successor times to arrive at the predecessor late start time. Figure B-5 illustrates the final version of the PERT/CPM chart with both the early start times and late start times calculated.

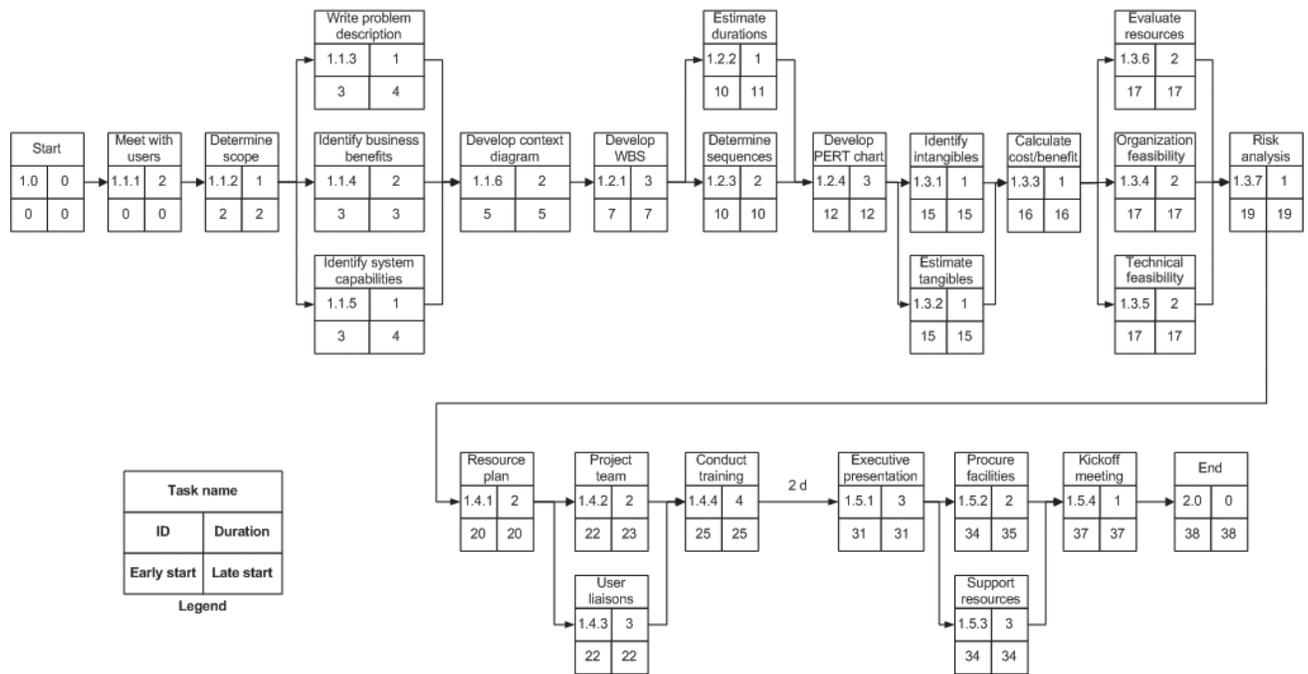


Figure B-5
Final PERT/CPM chart for RMO

slack time
the difference between the early start time and the late start time for a task; the amount of time a task can be delayed without delaying the project

SLACK TIME AND CRITICAL PATH

As we look at the chart in Figure B-5, we notice that some tasks have early and late start times that are equal, but in other tasks they are not equal. The ones that are not equal always appear on paths in which multiple concurrent tasks run in parallel. The difference between the early start time and the late start time is called **slack time**. The first group of concurrent tasks has three tasks together: 1.1.3, 1.1.4, and 1.1.5. Because the three tasks must be done concurrently, and task 1.1.4 requires two days, the other two tasks each have one day of slack time. In other words, they can be delayed by one day and still not have a negative impact on the project.

Notice that some tasks in this diagram have an early start time and a late start time that are the same. In other words, the earliest and the latest that the task can begin is the same day. There is no slack time for that task. Tasks with no slack time are on the project's *critical path*, which Chapter 3 defined as the longest path through the diagram that also represented the shortest possible time for the project to be completed. If any task on the critical path slips, then the entire project will slip. Looking at the PERT/CPM chart gives another, more precise definition of critical path: It consists of any task with an early start time and late start time that are equal. Every project has at least one critical path. You will notice that sometimes concurrent tasks can both be on the critical path. Obviously, the critical path is important for the project manager because those tasks must be monitored carefully to ensure that the project does not slip. As a project progresses, other tasks may become critical-path tasks if they slip or increase in size.

NETWORK DIAGRAMS IN MICROSOFT PROJECT

A PERT/CPM chart is useful in building the original schedule because it enables the project manager to easily view dependencies. For project tracking, the tracking Gantt chart view is more helpful because it shows the project on a calendar backdrop. MS Project, which is the most widely used project-scheduling tool, supports both types of charts. However, MS Project has enhanced the Gantt chart view, and it is now often chosen both for building the chart and

for tracking progress. As you learned in Chapter 3, the data-entry Gantt chart view is an effective way to enter tasks with their durations, dependencies, and resources. The data entry view of MS Project does not show critical-path information by default, however. To view critical-path information, you must either modify the preferences or display the tracking Gantt chart view. This view is shown in Chapter 3 to illustrate task dependencies and critical-path information.

In this appendix, let's work through the same example as we did in Chapter 3, except from the PERT/CPM chart viewpoint. We start with the RMO schedule after we have entered all the tasks and their durations. Figure B-6 illustrates the Gantt chart view of the starting point. Note that the task durations are indicated by the bar lengths, but that they all begin on the same date, indicating that no predecessor/successor information has been entered.

Figure B-6

Initial RMO Gantt chart with no task dependencies

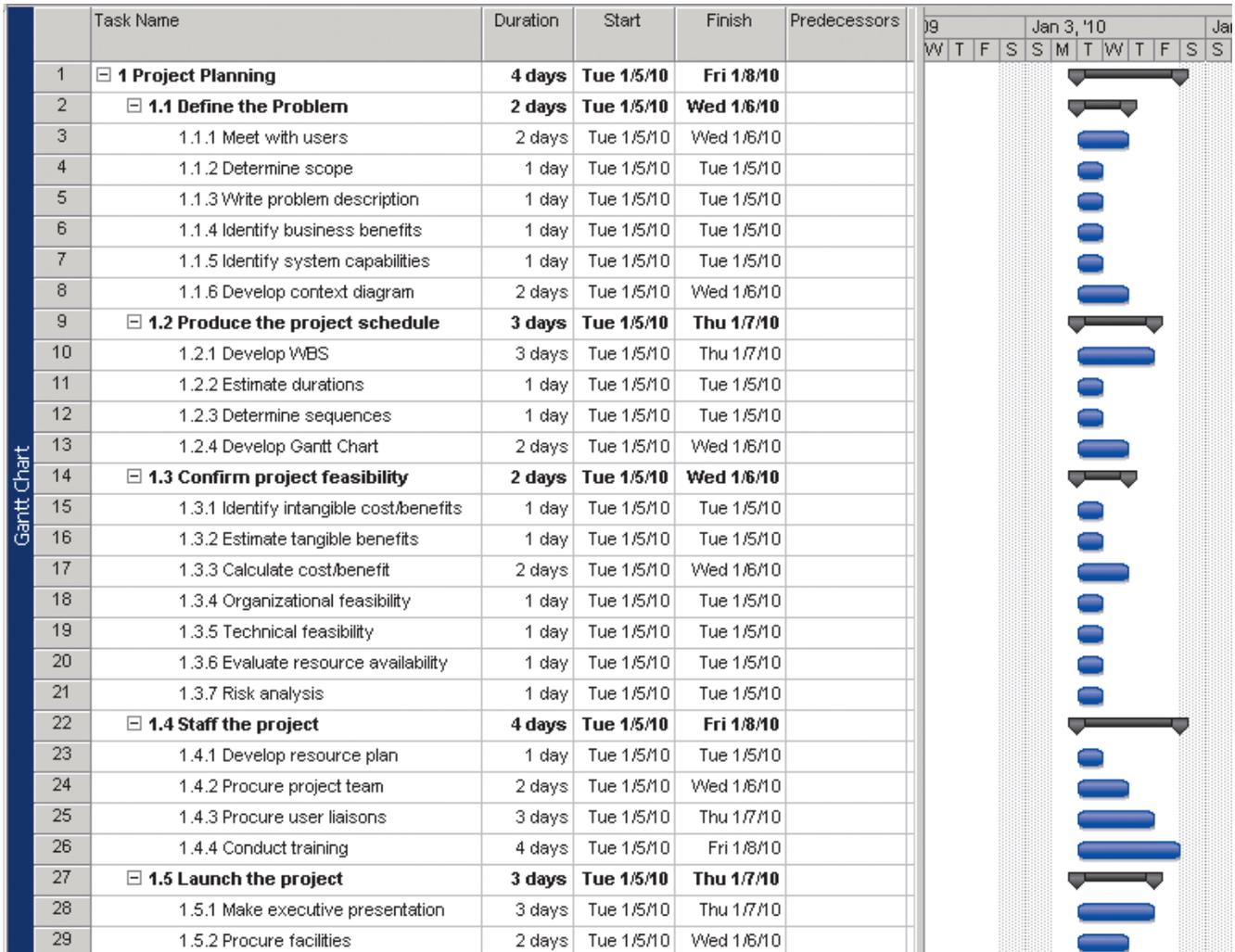


Figure B-7 shows the same information, except in network diagram view. MS Project calls its version of a PERT/CPM chart a network diagram. Because network view is a graphical layout of the tasks, it takes more space to view it all. Figure B-7 shows the first few detailed tasks as rectangle boxes and two summary activities as parallelograms.

You will want to modify two types of preferences to fit your own style. The Format menu contains preferences for Box and Layout. Box preferences allow you to change the information within each box. In our manual example, we should early start and late start so we can

learn more about critical concepts. In Figure B-7, we use the default MS Project box format, which shows start and finish dates. Note that the task IDs are assigned by the program and cannot be modified either in Gantt chart view or network diagram view. Layout preferences allow you to modify the layout of the boxes on the diagram. In the figure, the layout pop-up window shows the default values.

We are going to change three default values. At the top of the pop-up window, we select a parameter that allows us to rearrange the boxes manually. For purposes of this tutorial, we also deselect the parameter that permits the display of the summary tasks. This allows us to show more of the detailed tasks in the following figures. We also slightly adjusted the spacing values, again to allow more boxes on the screen shots.

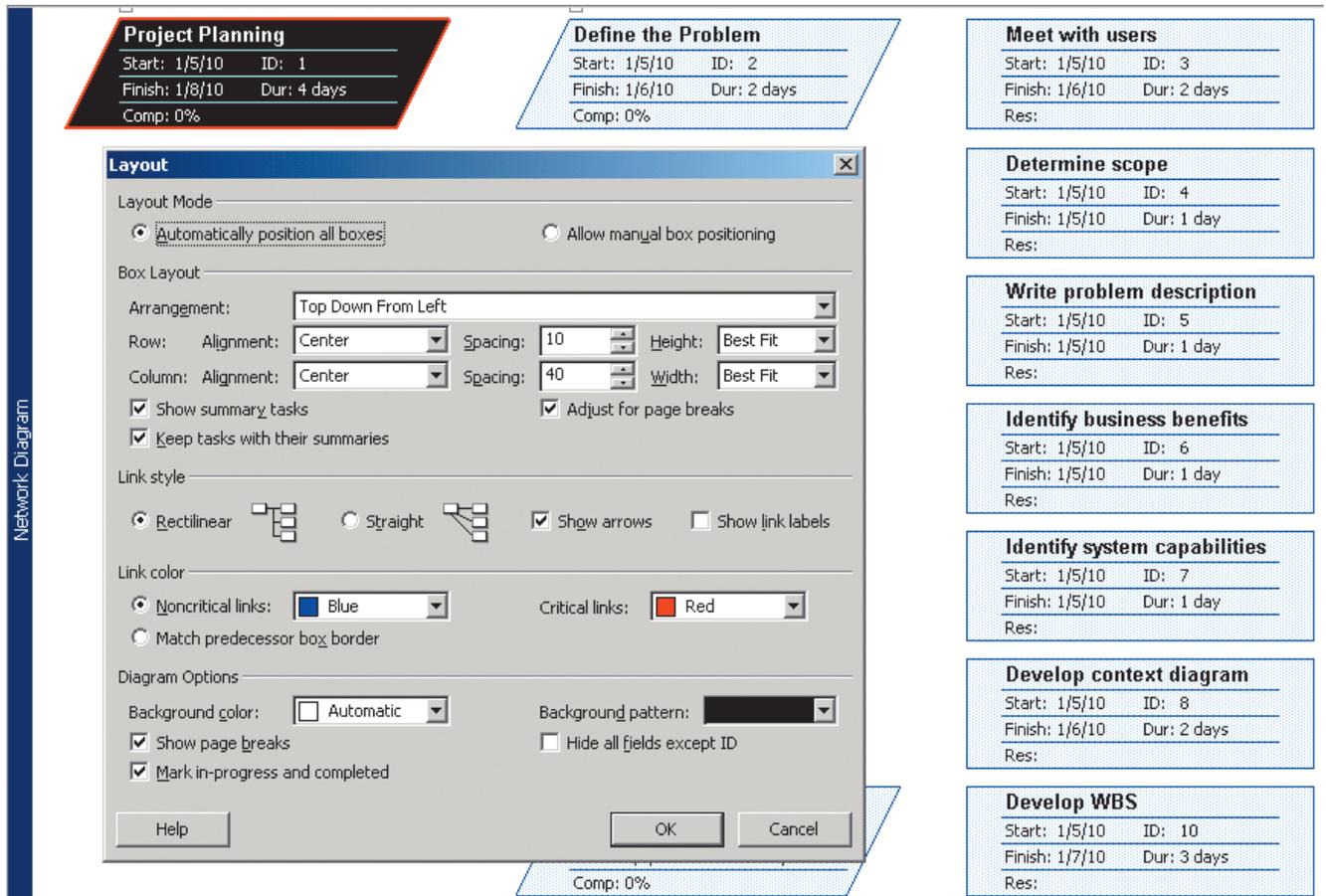


Figure B-7
Initial RMO network view

The next step in building the network diagram is to enter the dependency information. Network view has two methods that we can use. The first method is to use a split window, exactly like we did in Chapter 3. The second method is simply to drag connection lines between tasks by holding down the left mouse button and dragging a connection line from one task to its successor task. If you grab the border of a box, you can reposition the box. If you select the inside of a box, you can pull out a connection line and attach a successor task. Figure B-8 shows a network diagram with several task dependencies. The top half of the diagram has the dependency connections completed; those on the bottom half are not yet connected. As with the previous manual example, the visual layout of the project tasks makes it easy to note the dependencies between them. By default, MS Project also highlights the critical-path tasks in red.

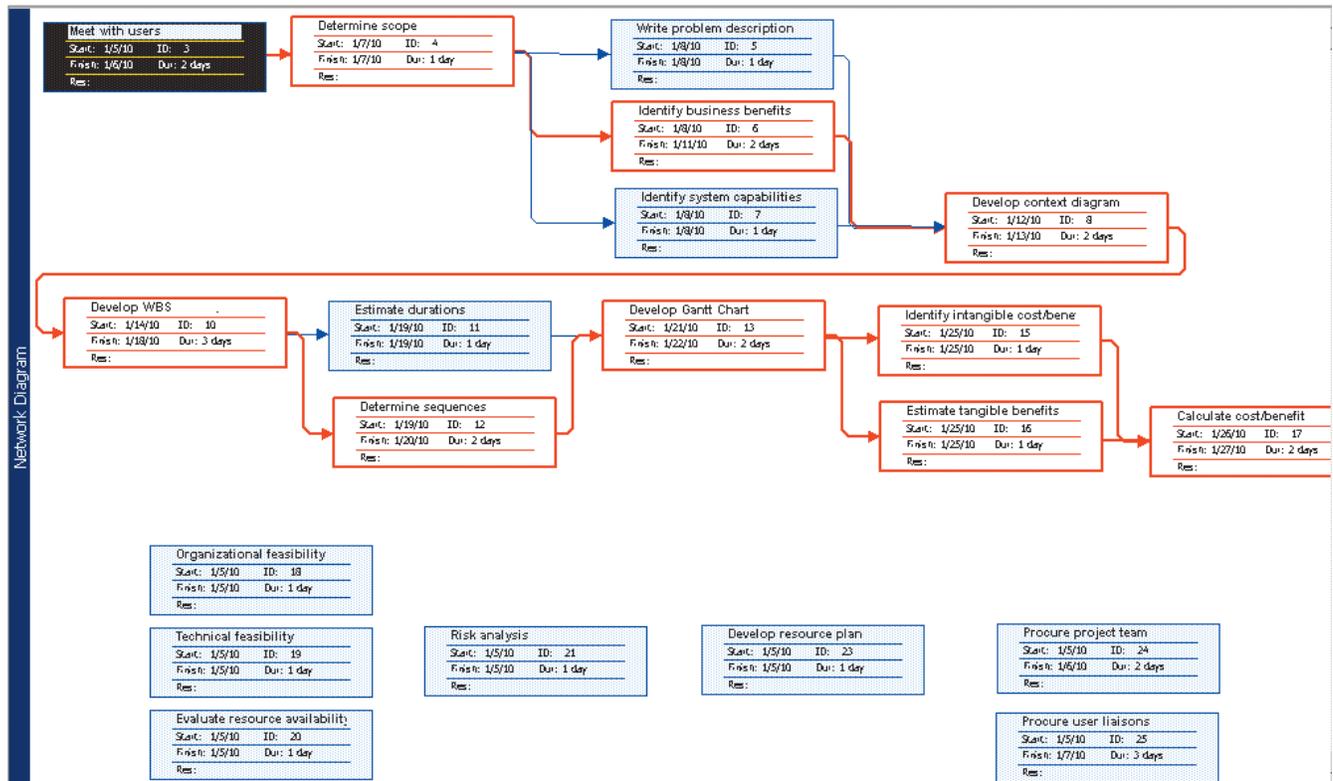


Figure B-8

Partial RMO network diagram with dependencies

Human resources can be identified and allocated to tasks in network view just as in Gantt chart view. Resources can be identified as individuals or simply as a type of employee, such as a programmer analyst. These resources can also be assigned a cost factor. Resources such as employees can be assigned cost factors by hour. Other resources, such as computers, can be allocated a cost per use or per item. Individual work schedules, such as full time, part time, normal workday, or normal week, can also be built. Project managers can generate various reports to show resource allocation—to check to see whether they have double-scheduled a resource or whether a person is underutilized. To enter resources for a task, first right-click the task and then select task information from the pop-up menu. Resources can then be assigned in the appropriate line, as shown in Figure B-9.

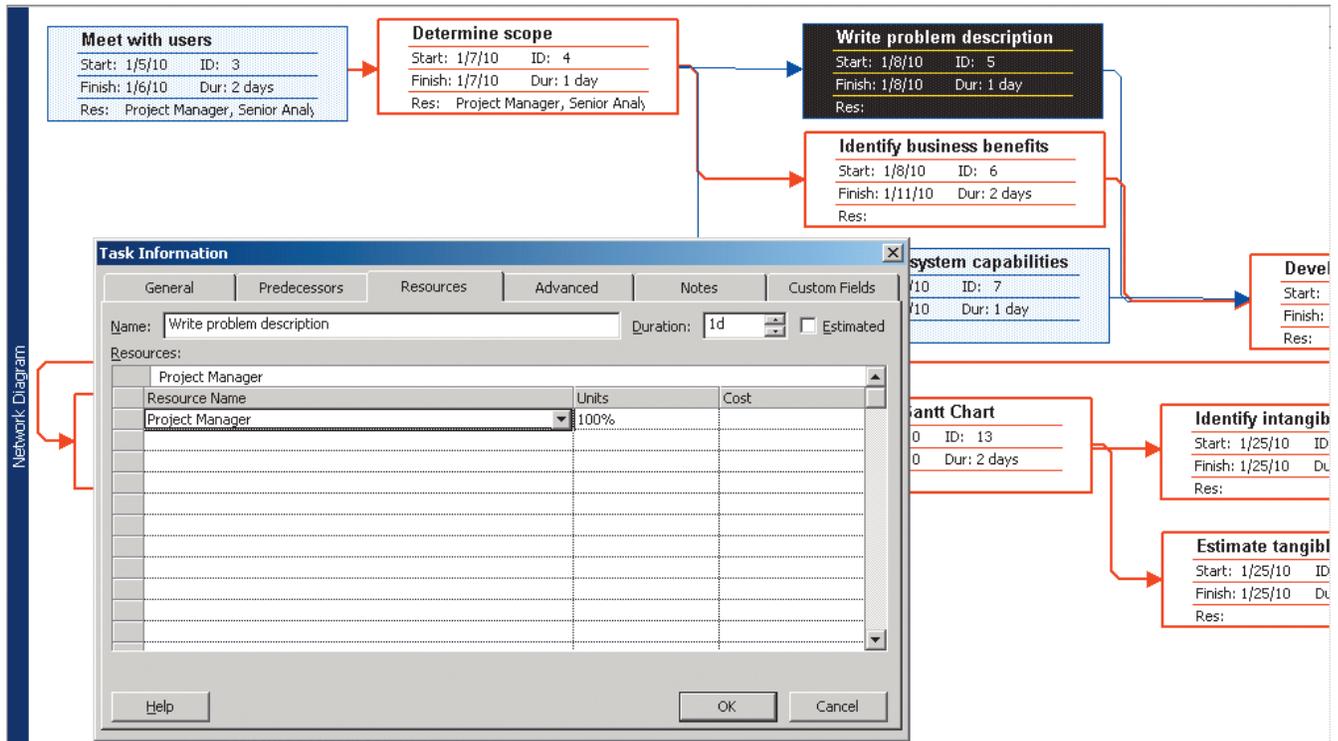


Figure B-9

Entering resource information into network view

Because costs are assigned to resources, the estimated costs of the project based on the schedule are calculated automatically when you assign resources. Various types of analysis reports are available, even earned value by task; you can see these reports by choosing different box formats. One of the more advanced features of MS Project is the ability to do resource leveling. As a project manager builds the schedule of work, resources sometimes are overallocated. In other words, they are scheduled to work more than a normal eight-hour day. The automatic resource leveling feature in MS Project will move the tasks to later points in the project to make sure that no resources are overallocated. It does so while maintaining the task dependencies.

TRACKING DIAGRAMS IN MICROSOFT PROJECT

Once all of this information is entered, it can be saved as a baseline schedule for later comparison. As the project progresses, actual completion dates and resource utilization can be entered. Figure B-10 shows a partial tracking Gantt chart view of the RMO schedule. Notice that each task has two bars. The project was first saved with a baseline, the estimate that was first developed. The gray bar, which is on the bottom half of the pair of bars, represents the baseline, and it is frozen. The top bar is the actual time that was required to complete the task. This figure shows increased time to complete task eight, *Develop context diagram*, from one day to two days. Note how the rest of the schedule is shifted to accommodate the increased time for a predecessor task to other tasks. In other words, all critical-path days are now projected to be one day late.

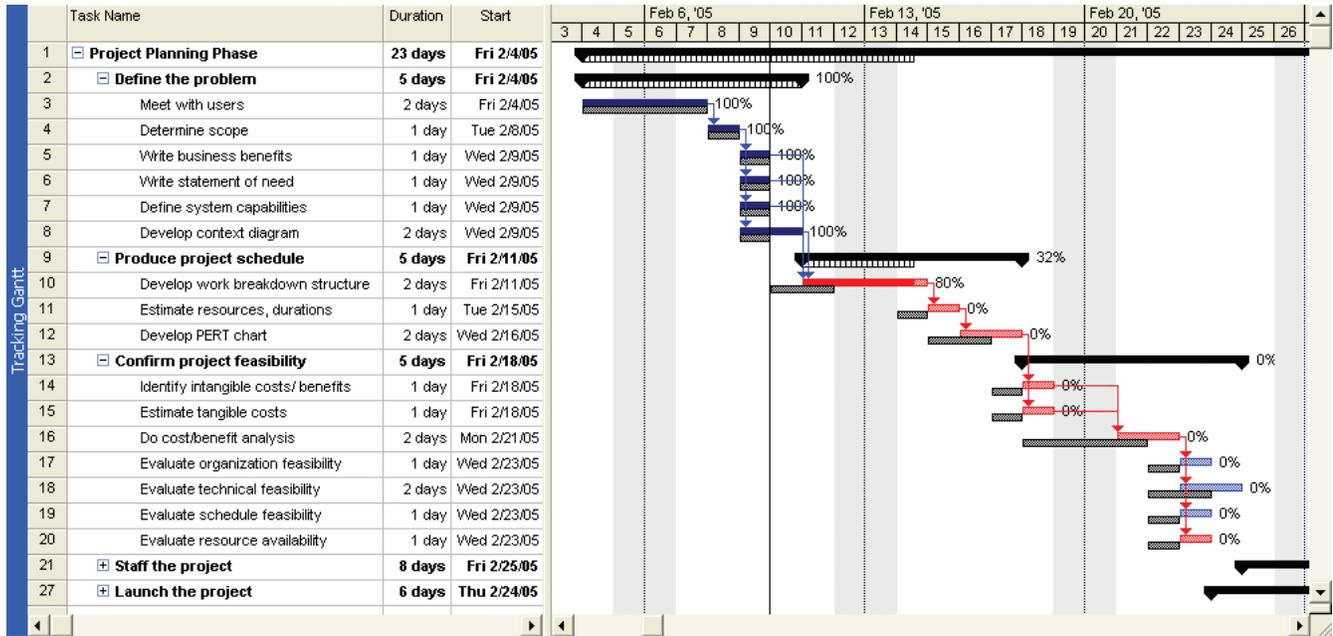


Figure B-10

A partial view of a tracking Gantt chart for RMO

The heavy vertical line represents the current date. Tasks to the left of the bar should be completed on schedule; tasks to the right should still be open. Tasks to the left that are not completed are behind schedule; completed tasks to the right are ahead of schedule. Uncompleted tasks that are on the critical path are shown in red.

Even though MS Project cannot determine automatically how the project should progress as described in the task durations and dependencies, you should be able to see how useful it is for putting the schedule together and for tracking. MS Project can help in other ways to manage a project. For example, it can track resources assigned to tasks and indicate when resources are overallocated. Estimated and actual project expenditures can also be recorded. The numerous reports and views that can be generated provide additional tools to help a project manager analyze and understand the progress of the project.



KEY TERMS

early start time, p. 5
late start time, p. 6

slack time, p. 7

REVIEW QUESTIONS

1. Why is it important to develop a detailed list of tasks when preparing to build a project schedule?
2. Explain in your own words the process of calculating the late start times on a PERT/CPM chart.
3. Discuss the difference between a PERT/CPM chart and a Gantt chart. List comparative advantages and disadvantages of each.
4. Can a project have two paths that are critical paths? Explain why or why not. Give an example, if possible.

THINKING CRITICALLY

1. You have a younger friend who is getting ready for college. Make a list of all the activities you think she needs to complete to start college. How did you decide how detailed to make your list?
2. Expand the list into individual tasks. Build a work breakdown structure that shows activities and tasks.
3. Draw a PERT/CPM chart of the tasks listed in problem 2.
4. Draw a Gantt chart for the tasks listed in problem 2.
5. Draw a PERT/CPM chart that illustrates how to get ready for class in the morning. Identify the critical path and the total time it takes.
6. Recall one of your classes in which you had a research project with other students—a term paper, a programming project, or some other major activity. Develop a work breakdown structure and a PERT/CPM chart for the project. Because this is a team project, you should be able to have multiple concurrent paths. Show the critical path.
7. Illustrate the tasks for problem 6 on a Gantt chart.