WHAT IS PROJECT MANAGEMENT SOFTWARE?

Project management software is a set of software functions and tools specially targeted at supporting the management of projects. As such, the software attempts to provide planning, tracking, analysis, and output support. Today, project management software is viewed as a key element in our software kit—the differences from the past are features and ability to link to other software and employ the network. Project management software was initially developed in the 1960s and 1970s to run on large computers. In the early 1980s with the emergence of personal computers, several packages for project management on a microcomputer emerged. One was VisiSchedule (developed by the makers of Visicalc, the predecessor to Lotus 1-2-3 and other spreadsheets). The earlier project management software tools focused on scheduling and producing output. Analysis tools and aids were minimal. This observation shows that a primary use of such software is for output to the project team and to management. After all,
what good is the tool if you cannot use it for analysis or output? More recent versions of such software have added more help functions, outlining, more examples, a wider variety of output, an ability to handle large projects, increased accounting capabilities, and the ability to handle multiple projects. People also use spreadsheets to track costs and schedules, but this is not project management software.

Software for project management is different from other software such as word processing or spreadsheets. First, it is used less often than other categories of software. Second, few people use it—most see the output from the software. Third, it is possible to customize your use of the software far more than many other types of software. Fourth, it tends to be more expensive than some of the common types of software. Finally, because such software is not as popular, there are fewer seminars, classes, and books available. This brings us to our first recommendation—if you use project management software, pick a popular and known software package. That way, you will not have to explain and defend your selection.

You can spend from less than $100 to over $5000 for project management software. It depends on what you want to do and how much time you have to spend on it. What you get out of project management is basically tracking and graphical output with some analysis support. On the other hand, what you put into it includes the initial learning curve, initial use and setup of the software, time required to embed the software into your everyday or every week schedule, and the effort required to become proficient. We assume that you, like us, have only limited time and effort available to learn and use the software. In this chapter, we want to focus on evaluation, selection, and use of the software from this realistic view.

Table 15.1 presents a series of features and capabilities of microcomputer-based project management software. With your own use in mind, you can later rate these in one of three areas: those that are important; those that are nice to have and why; and those that may not give you much benefit. Clearly, a project management system should be able to input resources, tasks, dependencies, and milestones. It should then produce a schedule and other reports. Now consider some of the features in Table 15.1.

- Multiple projects—The software should be able to link projects together to get an overall picture.
- Graphs and reports—The system should support a variety of graphs and reports (including PERT and GANTT) in different formats with different printers.
- Filtering—Filtering is the capability to extract a set of tasks and milestones from a project for analysis and reporting purposes. Filtering can be based on specific resources, time periods, dependencies, and other factors. Filtering gives you some of the capability of a database—flexibility.
TABLE 15.1 Features and Capabilities of Project Management Software

Handle costs in terms of standard, overtime, and per use costs per task.
View several projects at one time.
Relate one project as a subproject of another project.
Relate several tasks or milestones in one project to another.
Allow several projects to share a common pool of resources.
Be able to compare planned versus scheduled versus actual schedules.
Allow for multiple calendars for resources and projects.
Support filtering based on various criteria to restrict which tasks are considered.
Support a variety of printers and equipment.
Support interfaces into and out of spreadsheet, graphics, database and word processing software.
Support flexible reporting.
Provide PERT charts based on different levels of detail.
Support different formats for GANTT charts.
Support user interface and screen customization.
Preview a report on the screen prior to printing.
Support outlining of tasks under general tasks.
Support resource leveling.
Offer manual or automatic updating of schedules (to allow for bulk data entry).

- Multiple user access—shared access to project files is available through the network.
- Export and import—The system should support a range of file formats for importing information and outputting data for use in spreadsheets, graphics, database management systems, or word processing. Importing is useful because it is desirable to enter data into a spreadsheet quickly and then to bring it into the plan (e.g., tasks can be entered in such a way).

All of the software packages listed in Table 15.2 cover the basic features. Software differs not only in capabilities, but also in the environment. Some work only in a specific Apple MacIntosh or PC environment such as IBM or compatible, whereas others are Windows, Windows/NT, or UNIX-based. These factors should be considered along with price in eliminating some of the packages. The differences between inexpensive, moderate, and expensive software often lie in additional features, networking abilities, and more functions such as cost and expense projection, and so forth. Some questions you should ask in software evaluation are listed in Table 15.3. This will be expanded later in the chapter. Note that the level of expertise tends to peak unless the person is willing to expend enormous effort to learn detailed additional features.

Future features of project management software will likely include more Internet access, support for integrating schedules across a network, and
TABLE 15.2  Some Project Management Software Packages

<table>
<thead>
<tr>
<th>Name of Product</th>
<th>Manufacturer</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeline™</td>
<td>Symantec</td>
<td>IBM/compatible</td>
</tr>
<tr>
<td>Project™</td>
<td>Microsoft</td>
<td>IBM/compatible</td>
</tr>
<tr>
<td>Instaplan™</td>
<td>Instaplan</td>
<td>IBM/compatible</td>
</tr>
<tr>
<td>Project Scheduler™</td>
<td>Seator</td>
<td>IBM/compatible</td>
</tr>
<tr>
<td>Mac Project™</td>
<td>Clavis</td>
<td>Apple Macintosh</td>
</tr>
<tr>
<td>Project Management™</td>
<td>Primavera Systems</td>
<td>IBM/compatible</td>
</tr>
<tr>
<td>CA-Superproject™</td>
<td>Computer Associates</td>
<td>IBM/compatible</td>
</tr>
<tr>
<td>Flow Charting™ (for graphs)</td>
<td>Patton &amp; Patton</td>
<td>IBM/compatible</td>
</tr>
</tbody>
</table>

improved links with other network based software such as groupware and electronic mail.

WHY USE PROJECT MANAGEMENT SOFTWARE?

People often use project management software to meet some very specific needs. The software must be able to rapidly produce project reports for meetings. It should track and reveal dependencies, and should be able to export information to word processing, database, or spreadsheet software.

TABLE 15.3  Software Evaluation Criteria

- Compatibility with software you currently use
- Ability to operate on a network
- Internet linkage
- Integration with other software
- Features and capabilities that you need
- Ability to run efficiently (not minimally) on your computer system
- Ease of use and ease of learning
- Availability of reference and training books on the software
- Ability to handle projects of different sizes and types
- Compatibility with your printer
- Readability of the software manuals; availability of tutorial
- Extent of on-line help for software
- Software has had several previous versions to shake out errors and improve performance
- Operates on the network environment in your organization
- Compatible with software other people are using
- Ability to produce reports acceptable to management
- Ability to operate within your work flow and business process
so that you can embed information in reports. Why use it? It makes life easier. It is easier to use project management software to produce GANTT-
and other charts than a graphics package.

Using project management software is also wise politically as long as
it is not taken to the extreme. The software is useful for the project team
in that it indicates stable use of a tool. It shows structure to manage-
ment. It also can get you out of arguments on issues such as “what happens if . . . ”. The software tool is then the expert analyst.

Project management software does not manage the project for you. It
has limitations. The critical path that the software identifies will not often
be the real management critical path. Some software is not conducive to
tracking and accumulating costs. It can compare schedules and do “what
if” analysis if tasks slip. The software is more like a specific spreadsheet or
database tool as opposed to a general purpose tool such as word processing.

Like everything else, the project management software can be abused
in several ways:

- The software is acquired and never used.
- The software is used as a drawing tool.
- The software is used for timekeeping and budgeting and falls under
  its own weight.
- The software project is too large and unwieldy so that using the software
  for the entire project in one piece is not feasible.
- The project manager gets too involved in the software and keeps
  reorganizing the project and exploring the software—the project
  suffers.

To avoid abuse, we will be giving you some hints in the set up and use
of the software. Our fundamental belief is that if you handle several projects
that involve over 100 tasks and several people over a time period of at least
three months, then it is probably worthwhile to use the software—providing
you realistically plan your own time.

---

**HOW DOES PROJECT MANAGEMENT SOFTWARE WORK?**

We know that project management software takes as input the structure
of the project and the tasks, resources, costs, and other input and then
updates the schedule. Many software packages allow for schedule compar-
ison as well as for “what if” analysis in addition to graphic and tabular
output to paper or disk of the schedule. These are the steps in using
the software:
1. Set up the basic schedule information: name of project file, name of project, project leader, text; input milestones, tasks, and dependencies between tasks (this is the project template mentioned earlier); input resources and relate these to the tasks.

2. Estimate the effort required for each task. This effort will result in a completed first pass or initial baseline of the schedule.

3. Periodically update the schedule by indicating tasks completed, started, delayed, and so forth, as well as changes in resources. This updated schedule can now be used as a basis for project meetings and for comparing it with the baseline.

4. On an as-needed basis, perform what if analysis using the software and data.

As you can see, the effort using this approach is limited. The schedule might be updated every other week or even monthly for periodic meetings. It is certainly not used every day. When, however, we add timekeeping and cost analysis to the scope of use, we have to enter much more data, verify more information, and generate more reports. This is overkill. There is enough to do in the project without generating busy work.

---

**LEVELS OF USE**

We can identify at least five distinct types of activities that relate to the use of the software.

**REPORTING**

In this application, data is input for the purpose of producing graphs and output for meetings. This is often less work than using a graphics software package. This is a computer form of "etch-a-sketch" where the software supports drawing but its scheduling features are not used and resources are not applied.

**TRACKING**

Here the project work and effort are periodically logged in terms of completed tasks. The system is used to track progress and compare it with the approved plan.

**ANALYSIS**

Resources are assigned to tasks. The system is used to perform analysis on moving tasks around, changing task dependencies, changing resources
and tasks, and then seeing the impact on the schedule. Many packages allow this analysis. The problem is having the time to systematically lay out alternatives and to then input these one at a time to see the results. Otherwise, many changes could be input and the results would be muddied by the cross-impact between changes.

**COSTING AND ACCOUNTING**

In this mode, costs are assigned to resources. This can be done as a fixed resource or based on how much of the resource time and effort is consumed. Most project management software systems lack flexibility in handling costs as well as interfaces into budgeting and accounting systems.

**TIMEKEEPING**

The hours and tasks worked on by each member of the project team are entered into the schedule on a regular weekly or biweekly basis. Updating can occur through the network, reducing the data collection effort by a single person. Reports can then be generated on schedule versus actual resources consumed, costs, and so forth. A company already has a payroll system. Putting timekeeping on top of this creates extra work. Moreover, because the input is not used to compute pay and produce checks, the validity and completeness of the information are suspect.

The additional time consumed is not just in working with the software. You have to carefully design what and how to use the software for these additional tasks. Remember, “The more you want of the software, the more it wants of you.”

---

**GETTING STARTED WITH THE SOFTWARE**

Using the software is discussed before evaluation because your evaluation skills and knowledge will be improved if you know the situation facing you. Obviously, the first step in getting started is to open and load the software. We recommend that you install all options of the software right away to avoid problems later. Now for the data. The data files should be stored in a separate directory that is backed up. The data files do not need to be in a subdirectory on the same directory as the software. You should probably keep the data away from the software anyhow. That will reduce the amount of virus checking and data compression you will be doing.

After you have loaded the software, you now need to set it up for operation. Some of the process is similar to that for word processing; other items are unique to project management. Some of the setup tasks are
- Defining working and default directories;
- Identifying printers—naming the two most common in your office in case one breaks;
- Defining colors for the screen and other appearance options.

Another option is the working calendar. This allows you to enter the holidays and days off. Often, the software allows this to be applied to individual resources. For each day you can enter the start time and end time. In general, you should maintain a standard schedule. Otherwise, you will have to update and verify several schedules.

There are also options in layout that are important and unique to project management. One is the format of a task. A task description contains X characters; resource lengths are Y characters; and so forth. There are options on what items are visible on the screen. Although each software package is unique, here are some general guidelines:

1. Keep the task description length to no more than 35 characters. This makes the description field long enough to be descriptive and short enough to be readable with other information.
2. Restrict the resource field that shows on the screen to 8 characters. If you use abbreviations for resources and do not assign too many resources to a task, this is sufficient.
3. Granularity—You can specify whether a character on the screen equals an hour, a day, a week, a month, and so on. Because you initially want to see the entire schedule, it is a good idea to select at least a week or month. You can later change it with a keystroke or two.
4. Many project management software systems allow for either project update after each operation or a batch update after you have entered or updated a lot of information. Use the manual batch update. This will make data entry faster because you will not have to wait each time while the entire schedule is updated. Imagine how long you have to wait with 200 tasks and an individual task change in task 48! When you have made all changes or entered all data, you can then command the schedule to be updated.

After making these changes, you will probably have to save the results so that they will be the default when you start the software up again.

LEARNING THE SOFTWARE

After the initial loading of the software, the tendency might be to read the manual. Do not succumb. Start a sample schedule that was included in the software to see how it was set up. Remember you are learning how it should be set up with the software as opposed to learning project management. Use the sample project to learn the keyboard and what
actions are triggered by specific keys. Some specific ones you need are
(1) help; (2) save the schedule; (3) update the schedule; (4) get the menu;
(5) get task detail for editing; (6) delete a task; (7) set and delete dependen-
cies between tasks; (8) insert a task; (9) save the task; (10) print a GAN\-\NT
chart. After working with the sample schedule, go ahead and establish your
own phony schedule. Follow the standard conventions and rules discussed
earlier in the book:

1. File name for the schedule should be of the form: XXXXYZZZ where
   XXXX is the name of the schedule, Y indicates the version (B-baseline, L-
   latest, etc.), and ZZZ an abbreviated form of date. The Y can be used for
   B, A (actual), and P (planned). For example, we might use SAMPTA15 for
   a sample schedule in a test mode with the date of October (A) 15. Enter
   the name of the project. Using long file names, even when possible, can
   be confusing when viewed through standard file manager software.

2. Enter at least five tasks (A, B, C, D, E) and one milestone. Use the
   standard abbreviations you have set up as well as the task numbering dis-
   cussed earlier. Save this schedule.

3. Now set up dependencies between the tasks: A precedes B; B precedes
   E; C and D precede E and go in parallel to A and B. D can be started two
   weeks after C starts. Save this schedule.

4. Enter resources in the standard abbreviated form discussed earlier.
   Assign the resources to the tasks. Usually this can be done by editing the
   tasks. Save this schedule. This is your template.

5. Now develop schedules for each task using fixed or as soon as possible
   (ASAP). Set the milestone at the end as fixed. This is a firm deadline. The
   tasks then have to lead up to the milestone. When you enter the duration
   of a task, keep with days or weeks, as opposed to hours or months. When
   you have completed these steps, save this schedule.

You have now created four versions of the schedule that have individually
been saved as separate files. You can produce reports for these tasks and
schedules. You should produce GAN\-\NT, PERT, and resource charts as well
as detailed task charts to verify what you entered. Verification using reports
as opposed to trying to read the screen forms is often easier and gives you
a different perspective after you have been staring at the screen to input
the information. This work is useful for several reasons. First, you learn
the software. Second, you have examples that are similar to that of your
organization for use as handouts and explanations in meetings. Again, when
you work with software you want to accomplish several goals at one time.

PROJECT SETUP

We are now ready to begin to establish some standard templates and
files that will be useful in working with the software later. What we will be
doing here is creating a standard resource file in which there are resources with no assigned tasks. We will also create a file with a work breakdown structure but with no resources. Then we will create a file with both the work breakdown structure and the task list. You will be using the rules and conventions discussed earlier. Again, we remind you to avoid outlining because the indentation of the tasks consumes space (and you only have 35 or so characters available).

To learn more about the software and to accomplish this more quickly, we suggest that you enter the tasks into a word processor or spreadsheet and then import the file into the project management software. To do this, enter two tasks into the word processor or spreadsheet and save it in the format specified by the project management software. Then start up the project management software and attempt to import the two task or two resource files. If this does not work, then you will need to read the manuals for the project management software. But in any case, you did not lose any data because you did not enter all of the real tasks or resources. Again, you should always test a function with a very small amount of data so that you do not destroy your work.

There are typically a number of fields or attributes that you can supply for a test (depending on the software). Some of these are as follows:

1. Scheduling: fixed time, as soon as possible (ASAP), as late as possible (ALAP). In general, the only things you want to fix are milestones that are immutable. All tasks should be scheduled on an ASAP basis for flexibility. If, for example, you were to schedule five tasks in a fixed mode and there was a shift in plans, all schedules for these tasks would have to be manually changed.

2. Duration refers to elapsed time of the task.

3. Effort refers to the effort required over the elapsed time. The effort can be more or less than the duration.

4. Status—Typically, this can be finished, future, or in process. You should put in tasks that have been completed. This will provide the team members and management with a better understanding of the software. It will provide continuity.

5. Percent complete—Usually, the software allows you to put in a percent complete for tasks that are in process.

6. Resources can be organizations, equipment, software, or any entity, and can be identified as follows:
   a. Short name—This will appear on the GANTT charts and so should be very short to allow for fit. We recommend no more than four characters.
   b. Full name—This can be very descriptive, but must not be ambiguous. Exact organization names should be used.
   c. Calendar for individual resources—Many people run afoul of this. They attempt to be fancy and create separate schedules for each
resource. Then when the schedule pushes out tasks in odd ways, you
don’t how it happened. In general, you should not fall into this trap.
Use a standard calendar for all resources.

d. Leveling—Resource leveling is the option that allows the software to
change the schedule to level the resource usage across the project.
Because this often has a substantial impact on the schedule, it should
initially be avoided. If you invoke resource leveling and later wish to
change the schedule, you often have to edit each resource to take
off resource leveling. Let's see what resource leveling does to two
tasks (A and B) assigned to the same person for the same period of
one week. If these have the same priority, it may put the first one
first and the second one in second place. Or, it may have the person
work half-time on each. It depends on how you have set the software
to do the leveling. If one task has a higher priority, it will go first.
Resource leveling moves tasks that have overcommitted resources.

7. Text fields. These are user definable fields. Candidates might be
person responsible for the task, risk level of the task, issues (indicate
the number of the issues), action items, and lessons learned.

Resources should include more than organizations and roles such as
management and project manager. They should include equipment, facili-
ties, and other items needed to perform the tasks. However, you should
only include those resources on an exception basis. Politically, you will use
this to highlight the need for these additional resources.

Two specific artificial resources are “critical” and “milestone”. Tasks
that you know are important are critical resources. All milestones are the
milestone resources. Because almost all software packages allow the assign-
ment of multiple resources to a task, this should present no problem.
The ordinary resources can be assigned as well. Using the filter based on
resources, you can now generate reports based on the resources: critical
or milestone. Critical is for the management critical path discussed earlier.

After you have entered all of the tasks or resources and have saved the
files, you should produce a set of standard reports to use in various meetings
and to explain to people how you set up projects. Again, you are addressing
several needs through the same work.

There is one more thing you need to be able to do—filter a schedule.
That is, for example, have the software suppress all tasks except those that
use a specific resource. You should do this for not only learning it, but also
because you will need to be able to show someone their tasks in the project.

Once you have the structure of the project, you can then select the
specific tasks and enter dependencies. This will be saved as a new file so
you will not disturb your general file. Attach resources to tasks and save
the results. Make sure that you input a minimum number of dependencies.
Dependencies should be major and occur logically and should not be based
on the same resources or some other reason related to how things are to
be done (as opposed to what is to be done). Save the schedule and produce
reports. You will probably make mistakes that can be detected from the
PERT chart for dependencies and the GANTT chart for resource as-
signment.

You now have a schedule without specific dates and times. Enter the
latest schedule status for the schedule to date and save this as a separate
file. Update this file two weeks later and save it separately. Use the software
to compare the two schedules.

---

**PRESENTING THE SCHEDULE TO OTHERS**

If this is the first time people in your project have seen this project
management software output, you need to get them used to the appearance,
structure, and some basic capabilities of the software. You also need to tell
them what its role will be and how it will be used. How do you do this in
an interesting way? Here’s a method we have used with some success. Invite
the people in the project one at a time to your office. Show them the
reports and hand them a copy. Then turn around and show them how you
can update the schedule and compare two schedules. You might want to
show them how to filter a schedule. You are not training them—you are
showing them that the capability exists and that you are on top of the
software. This should not take more than 20–30 minutes. If it exceeds this,
you are trying to show off or you are training them. During this period,
you also must explain how the software will be used. Once you have estab-
lished the baseline schedule, you can disseminate it through the Web and
server so that people can review it, and with proper security, apply updates.

**USING THE SOFTWARE OUTPUT FOR PRESENTATIONS
AND MEETINGS**

After reading this title, you might think that we are going to talk about
fancy fonts and color output. You would be wrong. We want you to use
black and white output that can be copied and 8-1/2 × 11-inch paper. We
suggest staying away from pretentious presentations that detract from the
analysis and comprehension of the project information. Our first rule in
the use of the output is to have a minimum level of output and then only
expand this on a very selective basis. There are several approaches:

**Level 1**

Project is basically on track; you are giving out routine information on
the project. Use only the GANTT chart. Show only recent major milestones
and current and near-term future tasks. Print out at most 100 items.
Level 2

There is a project issue at hand. It may relate to a specific resource or set of tasks. You should produce the GANTT chart along with selected detailed reports and charts. An example might be to filter on all tasks and milestones that are dependent on the late tasks. Another situation might involve a critical resource or department. A filtered project can be established using the specific resource. Another report that is useful is the task detail report, which contains all of the information on a specific task.

Level 3

The project requires substantial change. You have to explain the situation and propose a course of action. To explain the current situation you should develop a projected schedule that illustrates the effects of slippage and other problem areas. You should avoid too much detail. Otherwise, there is a danger of being involved in finite analysis of the problem and not the solution. After presenting the problem, present a work breakdown structure that is your solution in restructuring. Get an understanding of how this structure is different. Then jump to the new schedule with the changes. Explain the new schedule by comparing it to the old schedule.

Level 4

The project is in major trouble. Either a major change or termination of the project should be considered. You should always consider shutting down the project. We present three scenarios: (1) do nothing different and watch the resources go up in smoke; (2) invoke major change and restructuring; (3) kill and phase out the project. The first two schedules can be generated in a way similar to that of the previous level. For the killing of the project, you need to do more work. In addition to a new task plan, you need to develop a political explanation acceptable to the organization. You also need to develop a rapid decline scenario that is still orderly. Address what can be salvaged from the project and what can be learned from the project. We will discuss these items in more detail in a later chapter.

Using the reports for competitive advantage

A project management software package typically has a range of reports available. We have identified some of these already. A more complete list is as follows:
• Gantt chart
• PERT chart
• Task detail
• Resource histogram—this shows the usage of a specific resource over time.
• Cross tabs—tasks are rows and resources are columns. This shows which resources are required for the tasks.
• Status—the status of currently active tasks

But this is only part of the picture. You also can select the subset of the plan to be reported. This is where you can gain advantage. Be selective filtering on tasks you can produce reports and charts that help your argument and make your point. Here are some examples:

• A department complains that they are too involved in the project and that it is consuming too much time. You should filter by each department and produce histograms for each department. This should illustrate fairness or equity. You should also produce the cross tabs mentioned above.
• A department or manager wishes to delay tasks. You can filter on all tasks dependent on the tasks that they propose to delay. This will show the impacts of delay.
• Some people are not working on their tasks to the extent that is required. The same filtering approach can be used.
• People are confused by the project because it is large. You should focus on the Critical tasks (those labeled with the resource Critical).

---

**UPDATING THE SCHEDULE**

Now let's suppose that the project has to be updated. We have talked generally about this. It is time to get precise. Obviously, you are not going to call everyone on the team every day and then post the percentage complete to all active tasks. You will probably contact everyone once a week and obtain a status report. Tasks that are completed can be marked at 100% complete. New tasks that are started can be marked as started with the start date changed, if necessary. We recommend that you do not change each active task that is still in process unless there is a change or slippage. In collaborative scheduling where each person updates, managers must review the schedule. When updating the schedule, tasks are marked as complete and dates are entered for actual start and finish. The baseline schedule which was approved before is not changed.

**VERSIONS OF THE SCHEDULE TO SAVE**

After reading the preceding section, it is clear that you will be generating a number of schedules. In addition to the baseline schedule, there are
updates to the schedule. There is the last updated version. Filtered schedules are usually not saved separately, but instead are generated as needed. Furthermore, there are schedule projections for specific scenarios as outlined above. In the file naming convention discussed previously, the Y or middle element can be used to include planning, actual baseline, and so forth. The project name should explain the purpose of the analysis and what was done. Why should we save these versions? We may need to review these if management asks why certain decisions were taken. Moreover, if you save the reports in paper form, you will lose flexibility of analysis because there is much more information behind the report that has not been printed.

---

SOME HINTS—POLITICAL AND OTHERWISE

When reports are printed, many often include the date and time that the report was printed. It can be embarrassing if you did the report at 5:00 a.m. and it is noted by people in a meeting (because the time is printed at the top of the report). Here’s a suggestion. Prior to printing the reports, exit the software to the operating system. Change the system date and time to a more politically acceptable day and time, say two days ago at 6:00 p.m. In DOS this is done through the DATE and TIME commands. Similar commands are available for Windows/NT and other operating systems. After changing these, restart the project management software and print reports. Remember to change it back after you have finished generating the reports.

Do not attempt to be an expert in the software or attempt to show off your software skills. It may make an impression exactly opposite to the one you want. If you attempt to explain nuances and details about the software, people may think you are a computer nerd. You will be viewed as a techie as opposed to a manager.

Be involved. If you depend on a staff member to prepare the reports and do the analysis, you will lose credibility because you cannot answer questions. People will figure out who to ask and will bypass you when they have questions. What should you do? At a minimum we feel you should acquire some expertise in using the software. You should have been involved in the set up of the software. You should also be involved in data analysis. What can you avoid? Data entry, running reports, and some of the data collection.

Use the software on a regular basis. If you get rusty, you will have to relearn the software. This can be very time consuming and frustrating. Do some analysis using filtering and comparisons of schedules. Filter by milestones and see what is coming up.
Another hint is to know when to avoid using the software at all. We discuss manual methods later, but we can comment that if a project has little risk, is short in duration (less than three months), and does not involve many tasks or resources, it would be overkill to use the software except for practice and learning.

Do not take the schedule results for granted. Too often people become overly dependent on the schedule and software. The schedule is only as good as the input. If some tasks were put in as late as possible (ALAP), then the schedule will likely be delayed. This suggests that you need to audit the tasks for accuracy. You can do this by printing out the report for task detail. This is tedious work, but it has to be done. It is, in fact, similar to auditing the formulae in a spreadsheet.

---

**STANDARDIZATION**

A common question we are asked is whether an organization should standardize on a specific project management software package. In principle this appears to be a good idea. For shared information common software is essential. Unfortunately, real life experience indicates that some software is better suited for small to medium projects; other software is better for large projects. Another thing to consider is the set of responsibilities that are incurred with the software endorsement. You may then have to provide training. To avoid this, name a technical expert for the software. Additional manuals and materials may be needed. If you are endorsing the software, you should also identify naming conventions and some of the rules we have discussed in this chapter.

You do have to do something, otherwise, you may have redundant and incompatible software. A suggested middle ground is to endorse a specific, popular software package that runs on the most common type of PC. Develop guidelines as to when it should and should not be used. Concentrate on middle-sized projects. Large projects have a life (and death!) of their own. Small projects are too much effort. Provide guidelines on the conventions and rules for the software. Have people try to learn as much on their own as possible. You need to encourage self-sufficiency. If there are sufficient users, then encourage them to get together now and then and share experiences.

---

**EXERCISES**

Find one or more articles on microcomputer-based project management software system. Sources are *PC Week, PC Magazine, PC World*, and over 100
other magazines. Typically, a magazine will provide an updated review every year or two. Also, use the Web to search for project management software articles, products, and reviews.

Scan the article(s) by focusing on the comparison tables. Read through the criteria of comparison. If the article selects one system as preferred, analyze why they prefer this particular one. If the selection is based on price, you can downgrade the review because the price on the street is often less than the suggested retail price. Now accepting the review approach determine which is the easiest to use and which can handle schedules of 200 tasks or less the best. Also, consider which has the widest range of output and supports filtering.

You will see by performing this analysis that the same package is not preferred for all criteria. By reading several articles written over several years you will see the changes in criteria that occur due to improvements in hardware and software. In general, you will be wanting to select software from an established software firm that has provided several previous versions.

Next, go to a book store that offers computer books. Look for manuals on project management software packages. If there are many books on a specific software product, you can possibly infer that the software is popular, but also may be complex. If there are no books on the software, then you can assume that the software has not achieved a level of popularity and installed base to justify a book.

Without spending any money, look around the office and talk to your friends to find out if they have a project management software package. If they recommend one, there are several things you can do. First, you can attempt to use the software to test how it works. Even if the software is several years old, you will now gain exposure and limited experience. Second, find out how the system is used or not used. Often you will find that it sits on the shelf. When you ask why it is not used, you may be met with embarrassment because the people feel bad about spending the money and not using it. Do not ask why they do not use it. Ask questions like who attempted to use it; what types of projects was it used on; was there any training?

--- EXAMPLE ---

The project management software used in the project was capable of running on both Macintosh and PC platforms. There was, in addition, a database that linked to the project management software. The database contained the project information while the project management software served as a front and back end to the database.

Some observations from the use of the project management software are interesting. First, the use of export and import of data to and from
spreadsheet software was substantial. A popular approach was to produce a table of resources (rows) versus resource use over time (columns). This could then be employed as the basis for calculating cost-related tables such as earned value. A second observation was that although there was a requirement for a minimum set of conventions on the structure of projects, there was a great deal of flexibility in how individual managers worked with the project data that was downloaded from the project database. This data was read-only and so the main data was protected. Updating the schedules and database was controlled.

Each area within a division, such as final assembly and integration, had standard task names and structure at the area and process level below the area. However, each area manager could customize the tasks below the level of a process. The database tracked at the level of the process. This approach supported both standardization for management as well as flexibility to accommodate different types of manufactured products.

There were four different schedules for each type of manufactured product. One was the schedule for the customer; the second was for marketing. The third was for operations management. The fourth was for the area manager. Multiple schedules allowed for analysis, data control, security, and yet ensured consistency. By the way, these replaced over ten existing schedules.

SUMMARY

Project management software can be very valuable for both management and political reasons. However, it requires substantial dedication of time and focus to successfully implement and use the software. If you think that you will not have the time or inclination to devote to the tasks we have identified, then don’t bother. You have to develop the pattern of using the software as a familiar tool, not just something on the side. But remember, you will have to come up with reports and analysis some way.