Practical Project Management – Tips and Traps

Part Two – Scheduling – Tips and Traps

Note: This summer, I celebrated my 40th anniversary of project management involvement with the release of my 2nd book: "***Practical Project Management: Tips, Tactics, and Tools" by Harvey A. Levine, John Wiley & Sons, 2002.*** Scattered throughout this text are some hundred or so Tips and Traps that are based on my experiences during these four decades of project management practice.

This is the second of a series of articles built around these tips and traps statements. It captures the essence of almost 400 pages of practical project management advice, in just a few short articles. Of course, you are invited to delve deeper into the material at a later time. Also, you will find selected excerpts of the book's chapters on this website as separate papers.

Critical Path Scheduling

TIP – Start without overlapping tasks

All critical path software packages provide the capability to overlap tasks, by applying start-to-start (S-S) or finish-to-finish (F-F) relationships.

A practical way of working with critical path scheduling is to start off by defining most relationships as default finish-to-start (FS) dependencies. Then, after the project schedule has been calculated, use the ability to overlap tasks to selectively compact the project duration. (Ch. 3-1)

TIP – Float and Slack are the same

The terms Float and Slack are used interchangeably. They are the same. Float had been the common usage, until Microsoft introduced their scheduling products, which substituted Slack for Float. (Ch. 3-1)

TOOL TIP – Suppressing Late dates or Early dates

Sometimes the user does not wish to publish the late dates or the float. This is controlled in the reporting process. In such situations, it is also popular to change the name of the early dates to something like **Scheduled Start** and **Scheduled Finish**, or perhaps just **Start** and **Finish**. Just about all programs allow the user to rename the standard calculated fields, both in reports and screen forms.

Also, at times, the user does not wish to calculate or publish the early dates. Instead they want to have the dates calculated and displayed as the latest dates. This would be equivalent to a just-in-time scheduling approach. Most products have an ALAP (as late as possible) calculation option that can be used to accomplish this. The default is the ASAP (as soon as possible) mode. (Ch. 3-1)

TRAP – Egregious overuse of imposed dates

First time users of project management software often tend to overuse imposed dates, especially the Start On option. In an attempt to force the schedule to predetermined dates, this improper use of the Start On option creates two problems. First, it prevents determination of a schedule that is based on defined dependencies. Second, it makes updating the schedule much more difficult, as the user must go in and manually change all of the imposed dates.

Often, this overuse of imposed Start On dates is motivated by a desire to avoid the effort of defining all of the dependencies. Ironically, the result is not only a poorly developed schedule, but also vastly increased effort to maintain it.

Furthermore, one of the great benefits of project management software comes from using these tools to help develop a supportable schedule, based on defined work, dependencies, and available resources. Ignoring all of this to create a forced schedule may be easier to do and more politically acceptable. But if it is not supported by the facts, what good is it in the long run? (Ch. 3-1)

TIP – Using "dummy" tasks to signify an external constraint

Another (unjustified) reason for imposing forced start-on dates is as a means of controlling (delaying) a task start due to an external constraint that is not defined in the project planning documents.

Creating dummy tasks to note such constraints is favorable over just imposing a Start-No-Earlier-Than (SNET) date on the affected task. There are two reasons for this. First, by using a distinct task, there is a specific, separate line item in which we can define the specific constraint. Second, there could be more than one such constraint for a task. We can use a separate constraint task for each. (Ch. 3-1)

TRICK – Multiple baselines

It is often desirable to be able to save multiple baselines. The first is usually the initial or contract baseline. A second might be a set of negotiated revision dates. I usually reserve one baseline set for my last schedule computation. Then I can compare the next update to that baseline to analyze changes during the last period. (Ch. 3-1)

Shared Contingency

TRAP – Don't bury schedule contingency in the task duration

Do you practice the concept of adding contingency to task durations? In most instances, we muck up the schedule by burying this contingency in the basic time estimate.

Schedule contingency is a vital component of a successful project. However, this contingency must be clearly identified and managed. Inconsistent and unstructured padding of time estimates, while a common practice, is not a good thing. There are better ways to allow for the uncertainty that exists in all projects. (Ch. 3-2)

TOOL TIP – Apply "shared Contingency" concepts

A new concept in critical path scheduling emerged in the 1990's, called "Critical Chain Project Management (CCPM). I have discussed and critiqued CCPM in an earlier paper. CCPM provides a method for improved management of schedule contingency, using a "shared contingency" approach.

Sciforma's latest scheduling release, **Project Scheduler 8**, is the first traditional CPM program to offer support for CCPM as an available option within the basic product. PS8 offers a complete Critical Chain capability as a scheduling option within the basic program. The CCPM capability in PS8 has support for multi-project critical chain scheduling, based on project priorities and constrained by key resources (Drum Resources feature). (Ch. 3-2)

Estimating Task Durations

TOOL TIP – Using "lags" between tasks to schedule a delay

With any of the CPM tools, it is possible to set a lag between the end of one task to the start of a successor. For instance, to add three days between Task A and Task B, we would define the link between these two tasks as "FS3". Task B can start 3 days after Task A finishes. In reality, the start of Task B is not actually delayed. It is just the schedule that will reflect the time allowance that has been inserted. (Ch. 3-3)

TRAP – Averaging task durations is dangerous

Be careful not to improperly use task duration averaging. For instance, we would not want to average performance on parallel paths. Let's say that we have Tasks A, B, C & D, each estimated to take 20 days. A, B & C actually take 15 days each. Task D actually takes 35 days. While the average still works out to 20 days, the actual duration for the path (for the four parallel tasks) is 35 days.

For another example, we look at two serial tasks, each estimated to take ten days. Task A gets done in 8 days. Task B takes 12 days. The chain probably took 22 days (rather than 20) because Task B didn't start until the 11th day. (Harvey's Law: A delay in one step is passed on to the next step. An advance made in one step is usually wasted). (Ch. 3-3)

TOOL TIP – Use the PERT method for uncertainty issues

An excellent way to address the issues associated with uncertainty (including contingency and risk) is to use the three time estimate approach, commonly called the "PERT method".

Special software is available to support the PERT method of schedule computation. This software supports both the three time estimate method and the defined-range approach, and provides statistical analysis of the confidence in meeting any date.

Some other products have provisions for using three time estimates, but do not provide full statistical calculation of schedule probabilities. Sciforma (PS7 and PS8) provides a less complete version of the PERT method, but (as a bonus) allows the user to vary the weighting of the three

estimates. With this capability, the user can calculate a range of possible project completion dates or confidence factors, by varying the weight of the three estimates. (Ch. 3-3)

TRAP – Remove contingency from task estimates to motivate schedule performance

The time to complete a task will almost always take a minimum of the allocated time, and probably more. If pressure is to be maintained to minimize the time spent on tasks, it is advantageous to move most contingency out of the individual tasks and allow for the contingency in other ways. (Ch. 3-3)

TOOL TIP – Use Hammocks to span a group of tasks

Hammocks are a very useful feature to automatically calculate the time between two points of the critical path network. Sometimes called "span" activities, they span the duration between the two designated points.

Use hammocks to determine the duration, resource usage, or cost on any task or group of tasks that span a variable time. For instance, to calculate the cost of a mobile crane that is needed for a set of activities. As the duration for this set of activities changes, the time and cost for the crane change with it.

The "hammock" feature is not universally available in project management software products. Hammocks are available in Sciforma's PS series of products. (Ch. 3-4)

TIP – Use date constraints (with caution) to impose earliest starts and required completions

The two most useful types of date constraints are: Start-No-Earlier-Than (SNET) and Finish-No-Later-Than (FNLT). The SNET constraint is used to control the earliest start of a task. It can be overridden by the CPM calculation. The FNLT constraint is essentially an imposed required completion date.

You can use the FNLT function to incorporate milestones from the Project Milestone Schedule into the detailed CPM. In fact, one can actually start with the PMS, setting the milestones as FNLT dates and then building up the details using the PMS as a schedule framework.

Caution is advised to not overuse date constraints, in place of letting the CPM calculate early and late dates. Imposed dates are an "exception" to the normal CPM calculation. (Ch. 3-5)

TRAP – Too much float can be a problem. Control with SNET and ALAP functions

The default mode for CPM computation is As-soon-as-possible (ASAP). This assumes that tasks will start as soon as any constraints are satisfied. Sometimes, total use of the ASAP mode creates excessive time in the schedule. One way to control this is with selected SNET dates, an alternative is As-late-as-possible (ALAP). This sets the late start as the scheduled start, removing all float or slack from the schedule. This is a way of building a "just-in-time" (JIT) schedule.

Building a schedule with too much float, is as bad as not having enough float. It will appear to be unrealistic, and will tend to be ignored. The judicial use of the JIT options allows the development of a more practical and believable schedule. (Ch. 3-5)

TIP – Using dummy activities for contingency

One way to remove contingency from tasks but still allow for a cushion is to create dummy activities to hold the contingency.

When adding dummy tasks for contingency, be sure to mark each of these with a code that can be used to identify such tasks and to select such tasks for contingency monitoring reports. By recording the baseline duration of these tasks (part of the normal "set baseline" function), you can produce a variance report, noting all reductions of durations for contingency tasks. You can even create an exception report, selecting only contingency tasks that have reduced durations. (Ch. 3-5)

Harvey A. Levine, with 40 years of service to the project management industry, is founder of **The Project Knowledge Group**, a consulting firm specializing in PM training, PM software selection, evaluation & implementation, and PM using microcomputers.

He has implemented or enhanced the project management capabilities of numerous firms, often combined with the selection or implementation of computerized project management tools. Mr. Levine is considered the leading consultant to the project management software industry and is recognized as the leading expert in tools for project management.

He has been an Adjunct Professor of Project Management at Rensselaer Polytechnic Institute and Boston University. And has conducted numerous project management public seminars for ASCE, AMA, IBM, and PMI.

Mr. Levine is the author of the book "Project Management using Microcomputers", and has been published extensively in other books, periodicals and videos.

Mr. Levine is a past president of the Project Management Institute and the recipient of *PMI's 1989 Distinguished Contribution to Project Management* award. Recently, he was recently elected as a *Fellow of PMI*.

Mr. Levine has offices in Saratoga Springs, NY and San Diego, CA and can be contacted via e-mail at: LevineHarv@cs.com