Scheduling Best Practices

Proper Use of Milestones and Constraints

By Paul Brough, Vice President of Warner's Scheduling Group

Warner
Construction Consultants, Inc.
2275 Research Boulevard
Suite 100
Rockville, Maryland 20850-3268

301 670-9020
301 670-7977 fax
www.warnercon.com

pbrought@warnercon.com

Paul Brough has experience in multiple phases of project management, including estimating, bidding, scheduling, cost control, and field supervision of commercial, industrial and institutional construction projects. He has prepared and analyzed delay claims, and served as an expert witness on delays in various dispute resolution forums.

Mr. Brough's responsibilities involve the preparation, analysis and review of Critical Path based (CPM) construction schedules. That work typically involves the development of duration estimates, Work Breakdown Structures (WBS), and cost / resource loaded CPM networks for Baseline Project Schedules. He has frequently been retained to oversee the subsequent preparation of updates, and analyzes progress and resource utilization measured against the Baseline. He is proficient in the preparation of CAD based graphics illustrating complex work sequencing, and subnetworks for proposed changes (fragnets).

Warner's series of articles continues with this examination of a scheduler's use of milestones and constraints. Through this series of Best Practices articles we expect to pass along observations and insights, highlighting the best practices for schedulers and project managers alike. This article will examine both the proper and improper, as well as the effective, and sometimes deceptive, uses of milestones and constraints.

Milestones and constraint dates can be used in a variety of situations with varying affects on the schedule. To use them correctly will help avoid common pitfalls that less experienced schedulers encounter. Incorrect use of these tools often is the result of an incomplete understanding of the sequencing logic needed for the project in question. Both milestone and constraint dates are often improperly used to replace logic or activity relationship ties. This can distort the critical path, misrepresent float available to an activity, and cause inefficiencies in the process of updating the schedule. In order to avoid these mistakes and misrepresentations, and get the maximum benefits of their utilization, the following guidelines for these tools are outlined below.

MILESTONES

General Purpose

Milestones are zero-day activities that call attention to noteworthy events in the project schedule. They can represent a variety of significant events depending on the project being scheduled. They may indicate either the start or completion of a significant series of events. Milestones can mark major events such as when a building structure is 'topped-out', or when the building is enclosed and made weather-tight. A milestone might also represent a major turning point for the project, such as a phased completion date or a temporary occupancy date. Milestones are also effectively used when they are inserted into the schedule to show contractually significant points in time. These contractually significant dates could be the Notice to Proceed date, interim completion milestones, milestones relating to important dates for payment, or the project completion date.

How Milestones are Utilized

Milestones are inserted into the schedule logic as any schedule activity would be inserted. They should have predecessor and successor ties unless there is a clear reason to leave them open. With the careful use of a milestone, an open-ended activity can be clarified. For example, if there is an activity restricting the start of a series of activities but that particular activity is out of the contractor's control it may be scheduled as a start milestone without a predecessor activity. An example of this might be when an independent utility company completes its electric service to the project, establishing 'permanent power', thereby enabling power dependent work such as mechanical systems start-up, elevator installation, etc.

Choosing whether a milestone is a start or finish milestone is more obvious in some situations, such as the NTP date, but choosing the type of milestone effects the display of activity sequencing and should be based on sound reasoning. For example, do you want the milestone to represent the completion of the building's structure or the beginning of the MEP (mechanical, electrical, and plumbing) work?

In a purely functional sense, the difference in utilizing a 'Start Milestone' as opposed to a 'Finish Milestone' is largely cosmetic or symbolic – the choice has no effect on forward and backward pass calculations when applied correctly. The only functional difference is the type of logic relationship allowed with the predecessor or successor activities. For instance, a Start Milestone's successors are limited to two of the four relationship types: Start-to-Start (SS) and Start-to-Finish (SF), of which the latter is almost never employed, at least not correctly. A Finish Milestone's successors must be either Finish-to-Start (FS) or Finish-to-Finish (FF).

The Scheduler must be knowledgeable of how individual software applications treat the use of milestones, as this can vary slightly. Experienced scheduler's may find that, for instance, in Primavera Project Planner (P3), the process of inputting and displaying milestones can make a Start Milestone more efficient to modify, update and report than a Finish Milestone.

Another thing to keep in mind is that in schedule reports containing columns for start and finish dates, a Milestone will differ notable from a Task in that one of those columns will be blank for a Milestone. Less experienced users of schedules often find this confusing, and the Scheduler should be prepared to explain why such data omissions are correct and appropriate.

How Milestones are Misused

One of the biggest pitfalls encountered in using milestones occurs when trying to restrict a milestone to a particular date in the schedule, as opposed to being logic driven. This is discussed in more detail in the constraints section below. Ideally, milestones should be logically tied to related schedule activities wherever possible. This means that, at a minimum, a start milestone should have successor (i.e., 'Notice to Proceed), and a finish milestone should have a predecessor (i.e. 'Occupancy / Beneficial Use). Milestones for events such as 'Building Weathertight' should have both. This is important for float calculations which may be skewed by a constraint date on a milestone (also see constraints section below).

A frequent misuse of Milestones is simply not using them enough. Milestones can be an efficient scheduling tool, taking the place of tasks which take less than a day to complete. A seldom used practice we have found successful is assigning milestones for significant placements of structural concrete. Another good example is public inspection approval events. By using the inspection point as a milestone three positive results occur; 1) the inspection point is highlighted visually on the schedule, 2) there is a notice to the project manager to be reminded to schedule inspections with the appropriate authorities, and 3) there is a visible representation of the fact that the inspection approval affects many critical downstream activities, thereby placing suitable pressure on those charged with executing the work required for a successful inspection. Another subtle advantage is that the data entry for updating, or assigning actual dates to, Milestones is half the work of updating a Task – a Milestone requires the assignment of either an actual start or finish date, whereas a Task requires the input of both an actual start and finish.

A word of caution, however, is that some software, such as Primavera 5.0, has automatic reports that rely on the milestone activities to provide a project summary on one page or less. This 'feature' is one factor when determining how many milestones to use. If there become too many milestones, flags might be more appropriate, as they would not be included on this one page report limitation.

One last incorrect procedure we sometimes encounter is the practice of setting up tasks as milestones by assigning the task zero duration. In the aforementioned P3 application, this generates the particularly non-sensical result of the Early *Finish* date for the task being calculated as one day *before* the Early *Start* date.

<u>CONSTRAINTS</u>

<u>General Purpose</u>

Constraints are used to force activities and/or milestones to start or finish on a given date, irrespective of their logic ties. Constraints allow schedulers to represent conditions that cannot otherwise be molded within the normal predecessor or successor logic. Constraints may also be used for an activity without a natural predecessor or successor. The best practice is to tie all the activities to a predecessor or successor wherever possible, but if this is not reasonable or intuitive, a constraint date may be appropriate. For example, there may be a portion of a renovation project that cannot be started until the occupant vacates. The move date is both outside of the Scheduler's control and is a date committed to by others. The date is not changed by progress of other activities relating to this activity, it is only changed by reaching a new agreement with the occupant. Thus, it is fixed in time and would be a proper use of a constraint. An appropriate way to model this in the Schedule is to set up a Start Milestone for the move, and apply a Start-No-Earlier-Than (SNE) Constraint to that milestone. Details of such constraint types are described below.

Constraints may also be used to measure an activity's float to a milestone or task that is not necessarily in the longest path to project completion. This is used effectively with phased or interim turnover milestones. By creating a "phase complete" milestone and applying a Finish-No-Later-Than (FNL) constraint to its Late Finish, the float for activities leading to that completion milestone can be measured irrespective of the completion of the overall project. This can highlight a series of activities as critical in order to meet a contract milestone such as a phased completion date, while not affecting the rest of the activities on the critical path of the overall project. A similar example would be using a finish-no-later-than constraint to ensure a Plant's shut-down date is not missed. Using a constraint in this way causes the Late Finish dates of activities on which the shut-down depends to be governed by the date to which the milestone is constrained, rather than to the overall completion date against which they would normally be calculated. This therefore reduces the float values of those activities.

Types of Constraints and How they are Utilized

The table below defines, and explains the use of, constraint tools available to the Scheduler.

Constraint	**Effects**	**Utilization**
Start-No-Earlier-Than (SNE)	Constrains the activity or milestone from *starting* before the assigned date.	To model the impact of events or actions outside the scope of the Schedule. The example given above of vacating an occupied portion of a project is a typical application. It cannot be assigned to a Finish Milestone.
Finish-No-Earlier-Than (FNE)	Constrains the activity or milestone from *finishing* before the assigned date.	Same as above, except it applies to the Early Finish. Typical use for an FNE would be the utility company example above, where perhaps one would constrain the finish of the mechanical equipment commissioning activity so it couldn't happen until permanent power is energized. It cannot be assigned to a Start Milestone.

Start-No-Later-Than (SNL)	Causes negative float to be incurred when activity or milestone Early Start is driven past the assigned date.	This is rarely used, but might be applicable to seasonal work, such as winter paving in cold climates, where the start of paving would show negative float, and therefore be highlighted as critical, if it was projected to start later than the winter asphalt plant shutdown.
Finish-No-Later-Than (FNL)	Same as above with respect to the Early Finish date.	This is more commonly used, especially for projects with multiple, phased occupancy dates. Activity strings are highlighted as critical if they are driving a constrained item past it's assigned 'no-later-than' date.
Zero Free Float	Causes the Early Dates of the constrained activity or milestone to occur on the dates that exhaust its Free Float.	Often used to schedule 'just-in-time' deliveries of materials or equipment. This is essentially a shortcut for preliminary schedules where the procurement tasks are not fully developed, and the technique causes the delivery to be scheduled no sooner than it is needed by its successor activity. For instance, the procurement process for millwork may be completed earlier than the date where material is wanted on the site. Assigning this constraint would cause the millwork delivery activity to be scheduled so it completes just prior to its installation activity.

Zero Total Float	Assigns a total float value of zero, causing the Early Dates to equal the Late Dates of the activity.	This is often useful for the analysis of a schedule, as it can be used to pinpoint the string of activities that are driving the activity or milestone being analyzed. For instance, if a complex set of predecessors were leading to the start of a non-critical drywall installation activity, i.e., building enclosure, inspections, similar work in other areas, assigning a zero total float constraint would cause the total float of its driving predecessors to be zero, thereby making it easier to isolate and examine that activity logic.
Expected Finish	A constraint that causes the program to calculate the Remaining Duration in such a way that the "Expected Finish" date is met. (An 'Expected Start' constraint is similar to a 'Start On' constraint.)	These constraints should really only be used for projecting schedules, as a tool for determining what effect different completion dates would have on the schedule, or in definite deliveries or events, where the date is predetermined. This constraint allows the program to update the activity's original and remaining duration to meet a finish date. Expected finish constraints should not be used regularly as a part of schedule updates. It may be easily removed to return the activity to its original duration and this is why it is sometimes used as a tool for experimenting with different possible outcomes for a schedule.

Mandatory Start or Finish	Mandatory constraints force an activity not only to have zero float but also forces the logic of associated activities to hold the constrained date.	A mandatory constraint will override the logic of the schedule when performing a schedule update for the sake of holding the date in place. A normal constraint date can be overridden by a schedule update but a mandatory constraint date cannot. Mandatory constraint dates are not considered a best practices in schedule updating for this reason. Similarly, if used in schedule leveling, the sequences might become altered in order to hold the constraint dates firm. With mandatory constraints an activity would be frozen from moving beyond the date constrained, thus potentially not reflecting reality. The mandatory constraint date does not show the effect of late or early completing activities. A normal constraint date would show negative float (behind schedule) if it was late to complete when its predecessor and successor activities had completed, whereas, a mandatory constraint would instead freeze the associated activities from being completed. Effective use of mandatory constraints requires recognition of the implications of its use. Some schedulers try to show that the schedule can finish on time by constraining the end date and then leveling the schedule. This is not recommended unless trying to create a recovery schedule and the resulting activity durations and logic are carefully reviewed.

Start and Finish 'Flags'	Similar to Milestones, Flags represent key points in time	Flags are similar to Milestones, however, in some software they differ greatly. In P3, Flags can only have predecessors, whereas Milestones can have predecessors and successors. Further, Flags automatically status to reflect the earliest start date of the predecessor for Start Flags, and the latest finish date for the predecessors for Finish Flags. The use of Flags can be valuable for reporting points in time, especially if they have more than one predecessor activity as they automatically status themselves.

How Constraints are Misused

In addition to the misuses of mandatory constraints cited above there are other missteps schedulers may take in using constraints. Excessive use of constraints can result in difficulty understanding which constraint is driving the float on a given activity. The critical path may also become distorted, requiring a careful review of each schedule update to determine how the true critical and longest path(s) are represented. One way to alleviate the problem of multiple constraints, and therefore multiple critical paths, is to employ a technique that quickly can relax the constraints so the predecessor activities reflect float values they would incur with respect to the longest path. P3 has a feature that allows the design of 'global changes', one of which can be set up to extend assigned FNL constraints so that the longest path sets the float values.

Schedulers may use constraints as a replacement for logic, but too many short cuts in this manner are an ill-advised practice. A constraint should not be substituted for an incomplete understanding of logic, and if it used temporarily it should be replaced as soon as the logic becomes apparent. Thus, on a preliminary schedule, constraint dates may appear more frequently, but these should be replaced with the detailed logic that reflects the reality of the scheduled activities. Very inexperienced schedulers may actually schedule the project with constraints in lieu of activity logic. This later becomes a problem when updating, because each activity date (whether started or not) has to be updated individually or erroneous schedule reporting will result.

As mentioned above, constraint dates can cause the calculated float to yield more than one critical path. This can result in a misinterpretation of the longest path on the project. One would not then be able to look at the zero-float activities and expect to find the critical path. To remedy this, a scheduler may create two reports, one showing all constraints removed, and the other with all constraints included. More advanced users can manage constraints versus longest path reporting through 'global changes' permitted by some software packages. This entails using an intelligent identification system to filter the constraint activities, adding a set number of days of duration to push all constraint dates out beyond the completion date, and then globally changing them back by subtracting the same set number of days.

Conclusion

There are many ways to use both milestones and constraints in a schedule to help accurately reflect the progress of work on a project. The particular software used can help determine the quantity of milestones. The problem with these tools is that they both can easily be misused and even add confusion to the logical sequencing of activities. Careful use of these tools is encouraged but schedulers should not avoid their use altogether.

John K. (Kimon) Yiasemides and Marco A. Ferrante, both Managing Consultants with Warner, also contributed to this article.