An Introduction to the DCMA 14-Point Assessment Guidelines

Steven J. Montgomery, PMP

TENSIX
CONSULTING

tensix.com

Copyright Notice

This publication and its content are the copyright of Ten Six Consulting, LLC - © Ten Six Consulting, LLC 2019. All rights reserved.

Any redistribution or reproduction of part or all of the contents in any form is strictly prohibited. No part of this document may be electronically reproduced or transmitted in any form or by any means, electronic or mechanical, or translated into another language, without the prior written consent of Ten Six Consulting, LLC.

All referenced trademarks are the property of their respective owners.

Disclaimer

Every effort has been made to ensure the highest level of accuracy has been achieved in describing the DCMA 14-point assessment. Ten Six Consulting, LLC cannot be held responsible for any consequences that may arise from using this material or the knowledge obtained herein on any project, system or process. Readers are agreeing that the techniques shown are best practice guidance only and do not constitute specific direction for the projects, policies, processes and procedures of any particular entity.

Contents

TENSIX
CONSULTING

Chapter 1

THE DCMA 14-POINT ASSESSMENT

Abstract

The Defense Contract Management Agency (DCMA) provides contract administration services for the Department of Defense (DoD). Their mission is to provide oversight and deliver “actionable acquisition insight” to ensure both contractors and government meet their contract obligations. As part of this effort, The DCMA developed a 14-point schedule assessment, which is documented in the Earned Value Management System (EVMS) – Program Analysis Pamphlet (PAP), a Portfolio Management and Integration policy. This assessment provides guidance for the inspection and improvement of project schedules. As much thought and consensus formed this assessment it serves as a useful resource for understanding sound industry scheduling practices. Ten Six Consulting details and explains the reasoning behind each assessment in an effort to enhance one’s practical scheduling knowledge.

Introduction

Where can project managers look for guidance? Well, the Project Management Institute (PMI) helps project managers practice sound standards through a Project Manager Professional (PMP) certification and continuing education program. PMI also provides an Earned Value Professional (PMI-EVP) certification program. PMI supports the development and enhancement of project managers and Program Management Office (PMO) staff through its many publications, online web portal, and local chapters. But when it comes to schedule specifics, a helpful measure of the potential effectiveness of a project schedule is the DCMA's 14-point assessment. This assessment was developed to examine a schedule's quality and/or soundness. DCMA released the 14-point assessment in an effort to improve scheduling practices. Schedules that are judged high quality, as determined by the 14-point assessment guidelines, have a significantly higher probability of success. The 14-point assessment seal of approval ensures a well-built, and, therefore, practical schedule. Come along as we detail each 14-point assessment in support of enhancing scheduling best practices.

Background

The direct impetus behind the 14-piont assessment was a March 2005, memorandum by the Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (USD AT&L) that required Integrated Master Schedules (IMS) for all contracts above $20 million. Further, this memorandum mandated DCMA to develop and maintain guidelines for schedule evaluation and conformity. DCMA responded by producing a 14-point assessment as a means of schedule quality control.

Usage

The DCMA 14-point assessment provides a, DoD-wide approach to schedule analysis. Note that the 14-point assessment is particularly suited to inspect schedules supporting earned value management (EVM) analysis. Through the various phases of its development and the successful application to a plethora of projects the DCMA 14-point assessment has become a proven metric. Its guidelines are a measure of Integrated Master Schedule (IMS) adherence and improvement.

Note that the 14-point assessment is not necessarily a pass/fail quality criterion. Exceeding assessment thresholds does not necessarily indicate schedule failure. But the assessment review provides the catalyst for more discussion and analysis to better understand the schedule situation. So metric correction is not always required, but understanding and documenting the project situation is important. It is important for Project Managers (PM) and their PMO staff to understand that exceeding the 14-point schedule assessment criteria may result in questions from their DCMA team (or review committee) that may or may not require formal corrective action plans.

Policy Dissemination and Training

The DCMA 14-point assessment is a publically available policy located on the DCMA policy publication page. It is listed as policy 200-1 under ‘Portfolio Management and Integration’. Visit http://www.dcma.mil/POLICIES/ to obtain a copy or download directly at the following link:

http://www.dcma.mil/Portals/31/Documents/Policy/DCMA-PAM-200-1.pdf?ver=2016-12-28-125801-627

Sources indicate that the DCMA previously provided an on-line training course for the schedule 14-point assessment, but the links to this training were not maintained. And it is not known whether this training was publically available. The training was designed to improve the skill levels of the government technical workforce to determine whether a submitted Integrated Master Plan (IMP) is realistic and suitable for monitoring project progress. Again, the training website was not maintained, and other means of training government resources are not known.

Industry Incorporation

Meanwhile several 3rd party scheduling software companies incorporated the schedule checks into their software. Most notable are Oracle's Primavera P6 Enterprise Project Portfolio Management (EPPM) scheduling software, Deltek Acumen and Steelray Project Manager Analysis, which includes the checks with indicators for thresholds. These tools can perform the 14-point assessment to ensure that schedules are built to comply with the guidelines.

Ten Six Consulting Effort

Ten Six Consulting's interest in the 14-point assessment comes from our commitment to scheduling best practice guidelines, and to better explain sound scheduling practice to our customers. The study and/or presentation of scheduling guidelines provides a valuable and relevant teaching tool, as much thought and census form the backbone of these assessments. The broad acceptance of the guidelines also demonstrates the value of understanding the underlining principles. Ten Six Consulting training courses have long been based upon sound scheduling best practice principles.

To help our customers, we share this 14-point assessment guide to complement the Earned Value Management System (EVMS) and Program Analysis Pamphlet (PAP), Policy No. 200.1.

TENSIX CONSULTING

The Pathway Forward

What follows are 14 chapters on all the DCMA assessments listed below:

1. Missing Logic
2. Leads (Negative Lag)
3. Positive Lags
4. Relationships
5. Hard Constraints
6. High Total Float (Total Slack)
7. Negative Total Float (Total Slack)
8. High Duration Activities
9. Invalid Dates
10. Resources
11. Missed Tasks
12. Critical Path Test
13. Critical Path Length Index (CPLI)
14. Baseline Execution Index (BEI)

Each chapter provides an overview and in-depth look at a respective assessment. Chapters are written to be as self-sufficient and as modular as possible. So diagrams may be repeated between chapters, but each chapter has something unique to add to the scheduling conversation. The intent is to help schedulers to better apply the 14-point assessment to examine schedules and to enhance their general knowledge of scheduling principles and best practices. So, our hope is that this publication serves as a training guide for the novice schedule analyst, and, perhaps, also a reference guide for schedulers at all skill levels.

Chapter 2

SCHEDULE 'MISSING LOGIC' INSPECTION

Assessment 1 - Missing Logic Overview

This assessment inspects the schedule to highlight all activities that have no predecessor and/or no successor task. These activities are often referred to as "dangling" activities. The intention of this assessment is to examine how well (or poorly) the schedule is logically linked together. The number of tasks without predecessors or successors should not exceed the threshold of 5% allowable. The formula is as follows initially, but for this guideline and all others following, there afterwards always use only those activities in progress and not started for the calculation:

$$\% \, \boldsymbol{Missing\ Logic} = \frac{\#\ \boldsymbol{of\ tasks\ missing\ logic}}{\#\ \boldsymbol{of\ incomplete\ tasks}} \boldsymbol{x100}$$

Assessment 1 – Missing Logic In-Depth

The first criteria and a major metric in the 14-point assessment looks for 'missing logic'. Schedule logic defines the relationships between activities. The goal in scheduling a project is not to mandate a start and finish date for each activity; perhaps, with constraints, but to implement logic that allows the scheduling software to calculate each activity start and finish dates. These dates are calculated relative to its logical relationship to its predecessors and successors. 'Missing logic' results in a static schedule that does not react well to changing project situations; activity dates are not automatically updated. But schedules are meant to be dynamic: they should have the flexibility to adjust to the changing project scenarios. Schedule logic is the driver that determines how dynamic and flexible a schedule is. Let’s explore this assessment to confirm your schedule is not ‘missing logic’. This chapter investigates the ‘missing logic’ DCMA assessment as a measure of sound project scheduling.

Schedule logic, i.e. the relationships between tasks, describes the cause-and-effect dependencies between activities. A dependency in fact is the relationship between the (start or) finish of one task and the start (or finish) of another task. Again, the dependency reflects the cause-and-effect or logical relationship between two tasks. This logical relationship is detailed using the precedence diagram, Figure 1.

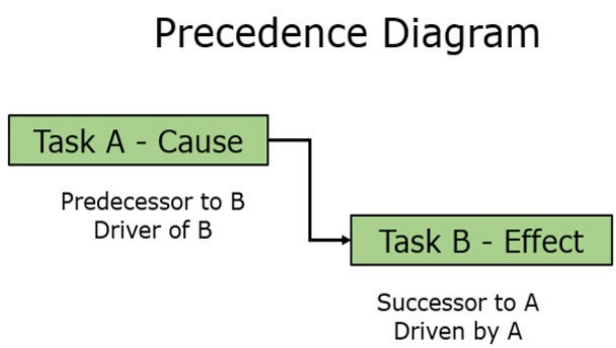


Figure 1

In this diagram nodes or rectangular bars represent activities, and arrows show activity dependencies. Also in this diagram, task A is the independent cause or driver and task B is the dependent effect or driven. There are four task relationships described by the precedence diagram:

1. Start-to-Start (SS): This is a relationship where one activity cannot start until another activity has begun. An example is that demolition cannot proceed until the safety plan has commenced.

2. Finish-to-Start (FS): This is the most common type of relationship where one activity can not start until another associated activity has finished. An obvious example is that an installation wall frames activity cannot start until a floor installation task is complete.

3. Finish-to-Finish (FF): In this relationship, one activity cannot finish until another related activity has finished. A simple example is that editing a report cannot complete until report writing efforts are complete.

4. Start-to-Finish (SF): In this not so common relationship, one activity cannot finish until another activity has begun. This relationship type is confusing and discouraged by most scheduling guidelines. The SF relationship is typically used by master schedulers and is a valuable tool in the proposal and initial planning stages of a project. Detailed planners then take the schedule and convert the SF relationships into more

understandable network logic by using the more common FS, SS, and FF relationships.

It is beyond the scope of this report to discuss the SF relationship in detail. But a practical application of the SF relationship might relate the Test Planning and Testing efforts early in the scheduling planning stages. In this situation, a certain portion of Test Planning must be done before Testing can begin. Testing cannot conclude until Testing Planning has been completed. A SF relationship between Test Planning and Testing adjusted by a waiting time helps to ensure Testing does not conclude until Test Planning is complete. This waiting time is lag, which modifies the SF relationship. The lag in this situation must be a sufficient value to make certain the Testing successor does not complete before its immediate predecessor, Test Planning. Lag is discussed in a later chapter.

Precedence diagrams of each relationship type are displayed in Figures 2, 3, 4, and 5 for your reference. Note that Figure 5 displays the SF relationship with a positive lag. Again, lag is discussed in a later chapter. It is included in the SF precedence diagram, as this presents the most practical application of the SF relationship.

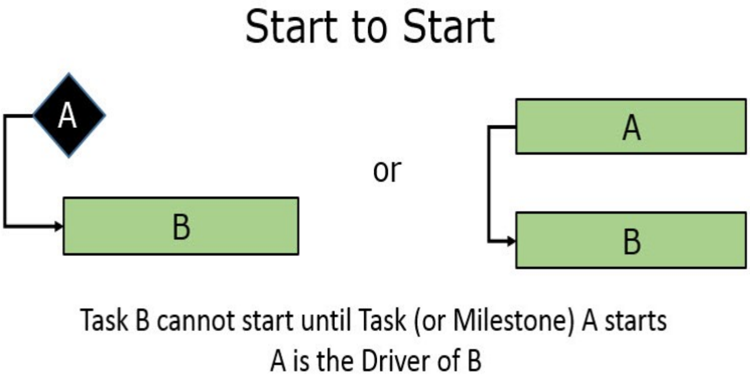


Figure 2

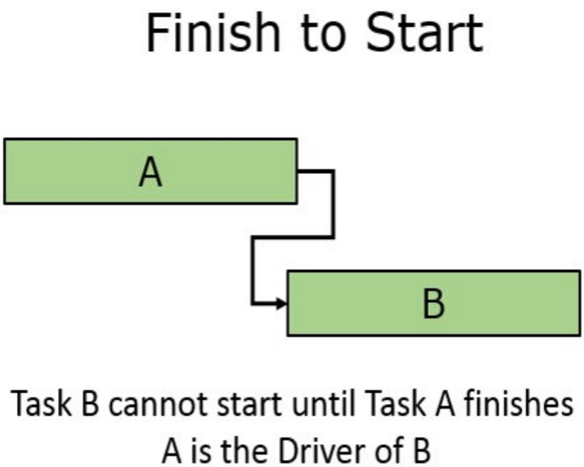


Figure 3

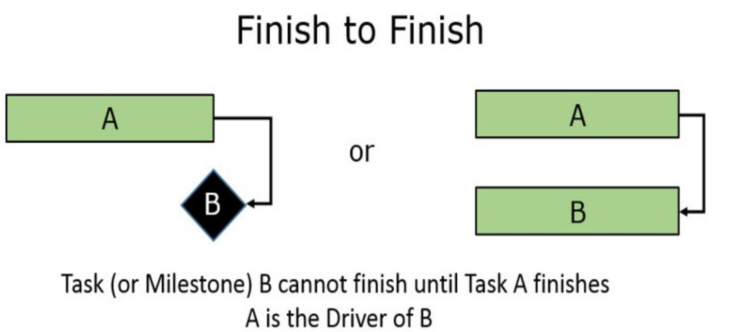


Figure 4

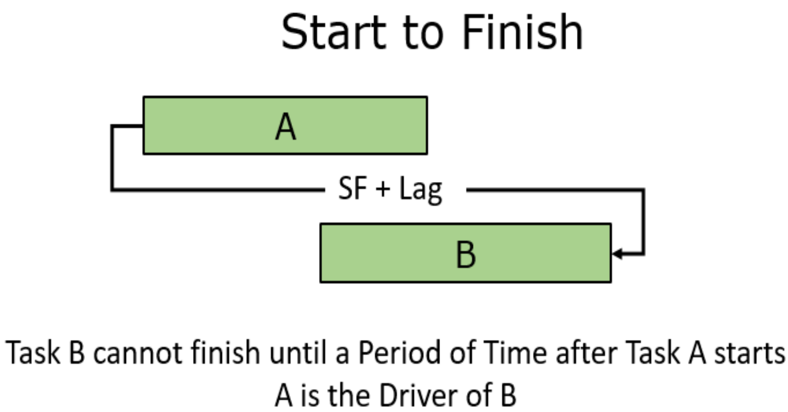


Figure 5

A common mistake is to think of dependencies as the chronological or time scale sequencing of activities. This is why we emphasize that dependencies describe the cause-and-effect relationship between activities. Think cause-and-effect or independent driver and dependent driven. This enables you to understand SS and FF relationships that appear to occur simultaneously. In both these relationship types the predecessor is the defined cause or driver and the successor is the resulting effect or driven.

Many times project managers have a particular date in mind when they schedule certain tasks, e.g. a date to mobilize equipment and crew. This leads them down the faulty path of mandating a particular start date for their activity using a constraint. You do not want to do this. Constraints add additional stipulations on tasks that make the schedule static and not dynamic. Insertion of numerous task constraints into a schedule is like putting a straight-jacket on the schedule. The schedule becomes inflexible and immobile or unable to adjust or react well to inevitable schedule changes. The imposition of contractual constraints in the schedule can identify critical impacts upon scheduled tasks and their associated logic that need to be addressed. But ordinarily, we do not encourage the use of constraints (other than the obvious overall planned contract start or completion dates) due to the adverse impact on the logical scheduling process.

Good flexible schedules have task start and finish dates based upon task estimated durations and the dependencies between tasks. This way when a task slips or extends the scheduling software is able to recalculate the start and finish dates of all tasks in the entire schedule. Good scheduler's therefore rely on scheduling software, along with realistic estimated task durations and sound dependencies, to compute task start and finish dates.

Another negative of inserting numerous task start and/or finish constraints in your schedule is that you lose your critical path. It is important to know your schedule's critical and longest path in support of schedule optimization efforts. You optimize your schedule by shortening durations or adjusting relationships of tasks along the longest path through the schedule. A schedule that has constraints instead of dependencies will be disjointed and not have a continuous longest path through the schedule. Not good!

This brings us to our topic of discussion; the DCMA 14-point assessment and 'missing logic'. The 'missing logic' criteria shows how well (or poorly) schedule tasks are linked together. The way to confirm a schedule is dynamic (or well linked and flexible) is to verify that in general every activity has both a predecessor and a successor (so no apparent dangling activities). Every rule has exceptions and in this case the caveats are the program start (has no predecessor) and program completion (has no successor). Also, although constraints are discouraged, a contractually defined constraint, e.g. contract completion date constraint, is acceptable and is normally part of the basic scheduling process.

The goal of the 'missing logic' assessment is to check all tasks for a possible missing predecessor, successor, or both. The assessment is adamant; the test fails if any activity (other than the start and finish of the schedule) doesn't have both a predecessor and successor. In other words, all tasks must be logically connected to pass. And task start and finish dates, therefore, should proceed from the schedule network logic. Task start and finish dates (with minor exceptions) must not be limited by arbitrary constraint dates.

Summary

A scheduler's major duty includes insertion of task dependencies, so task dates are driven by logic and not, again, arbitrary constraints. You want schedule logic to hold task dates in place, and not constraints.

The 'missing logic' assessment simply looks for tasks having no predecessor, successor, or both. Exceptions aside, this inspection ensures a logically driven dynamic schedule, i.e. a flexible schedule that reacts well or promptly to schedule updates. It also discourages scheduler's tempted to leave, e.g. 'pre-warranty conference', tasks dangling without a successor past the contract completion date. The 'missing logic' assessment mandates these types of apparent dangling tasks be logically tied into the completion of the project.

Chapter 3

NEGATIVE LAG (LEADS)

Assessment 2 – Negative Lag (Lead) Overview

Leads are logic relationships with negative lag. Leads are forbidden by the leads assessment. We will discuss the reasoning behind this prohibition, but suffice it to now say do not use leads in your schedule. When analyzing the schedule, tabulate how close to *leads free* the schedule is using the following equation:

$$\% \, \boldsymbol{Leads} = \frac{\boldsymbol{\# \, of \, logic \, links \, with \, negative \, lags}}{\boldsymbol{\# \, of \, logic \, links}} \boldsymbol{x100}$$

Again, the schedule should be free of leads, so this metric must compute to zero in the final approved schedule.

Assessment 2 – Negative Lag (Lead) In-Depth

The second check or criteria in the 14-point assessment looks for negative lag also known as a ‘lead’. Negative lags or leads are discouraged by the DCMA 14-point assessment because at best they can interrupt the forward flow of activities from beginning to end. And, if used improperly, negative lags can easily generate unrealistic network logic. This chapter examines in detail why the DCMA lead criteria forbids the use of negative lags and looks at possible alternative scheduling approaches. Potential resource conflicts due to leads, however, are not addressed.

First, let's be clear about one thing – this article is talking specifically about the DCMA 14-point assessment, which has a zero tolerance for negative lag in any defense schedule it is auditing. If you use negative lag, no matter what the rationale for doing so, your schedule will fail the assessment. However, use of negative lag in civil/non-defense contracts is performed at the discretion of the scheduler. And, while generally not recommended for the reasons spelled out in this article, is not an absolute rule – just a guideline.

Many schedulers prefer the use of negative lag in their schedules because it helps to compress the duration of the project. But during the schedule analysis the team must assess the practicality of the negative lag. Some schedulers also say insertion of negative lag helps to better model the true narrative of activities. However, the DCMA negative lag guidelines forbid the use of any negative lag in the schedule. They simply identify negative lag as a late item, which potentially could have an adverse impact on the schedule's critical path. As previously mentioned, any schedule that has negative lag fails this second check in the 14-point assessment. So if you are working on a defense contract that the DCMA is auditing – don't use it.

Let us return to the precedence diagram from chapter two and introduce associated positive or negative lag relationship modifiers. Again, the precedence diagram, Figure 1, describes the cause-and-effect dependencies between two tasks.

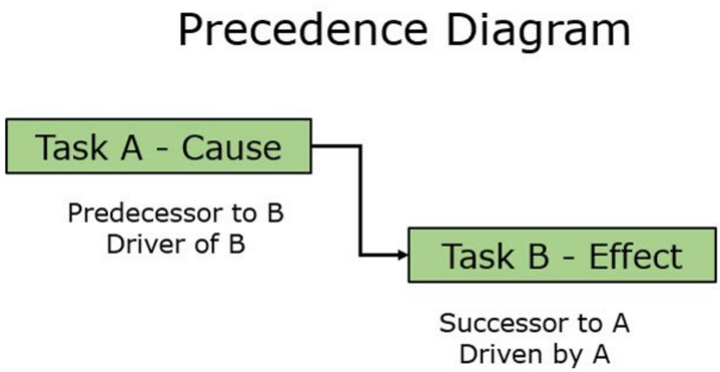


Figure 1

Remember that in this diagram task A is the predecessor or driver and task B is the successor or driven. Recall from chapter 2 that there are four relationships one may apply to model the interaction between tasks: SS, FS, SF, and FF. Refer to the chapter 2 missing logic assessment for a primer on the four relationship types. But note that the most common relationship and the one most often used with negative lag is the FS relationship.

It is possible to customize each relationship type with a lag modifier. This lag modifier is essentially waiting time or delay time. This waiting time modifier is referred to as lag. Figure 2 displays a FS relationship and positive lag modifier.

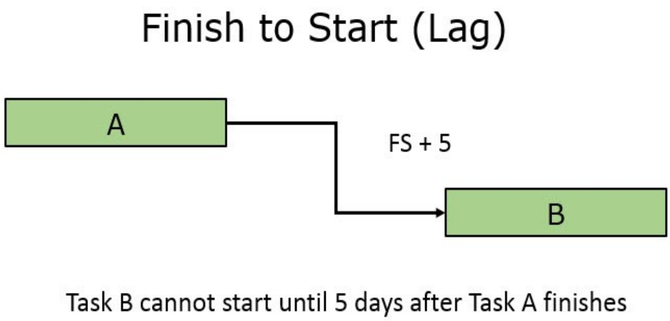


Figure 2

In this case the lag is 5-days. Here you must wait 5-days after the completion of task A before you can begin task B. Lag modifiers come in two flavors: positive lag and negative lag (or lead). Lead is negative time, which sounds a bit confusing. In Figure 3 we have our classic FS relationship modified by a negative lag (or lead). This situation reads “7-days before the finish of activity A you may proceed with activity B”.

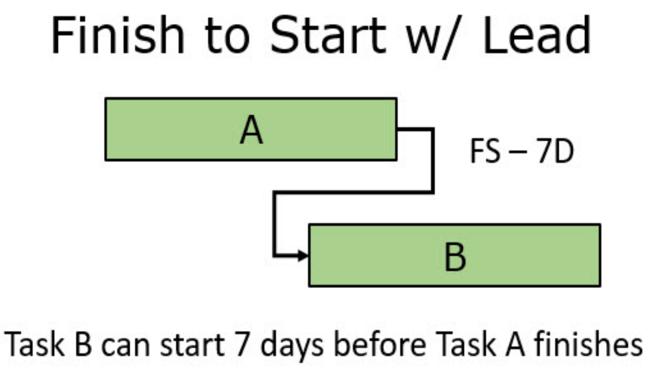


Figure 3

In this modified relationship you are saying that the FS relationship between activity A and activity B is not a hard set in stone relationship. The FS relationship in the precedence diagram says activity A must be completely done before commencing activity B. Well, the negative lag says that is not necessarily true. We can, perhaps, commence activity B a few days before the actual completion of activity A.

Schedulers like lead as it is a way to fast track the schedule by performing a portion of each activity in parallel. And sometimes a lead modifier seems to better fit the activity narrative. For instance the schedule that says 5-days before you receive equipment on site you begin mobilization. That appears logical. You know the delivery date, so 5-days before that date you mobilize your material, and then crew. It's a nice fit between mobilize and receive equipment on site.

But what happens if your delivery date is delayed? Oops! In that case you have your crew ready to proceed, but they are left waiting in limbo for the equipment to arrive on site. Not good! Thus the true story of negative lag is revealed. Negative lags imply the unusual measurement of negative time and require an exact foresight about future events.

Besides the required future foresight fundamental flaw with negative lag the DCMA 14-point assessment says that leads may distort total float and adversely affect the critical path. The DCMA assessment does not provide any examples of float distortion.

However, in Figure 4, we have a clear improper application of negative lag that not only distorts float, but also violates network logic.

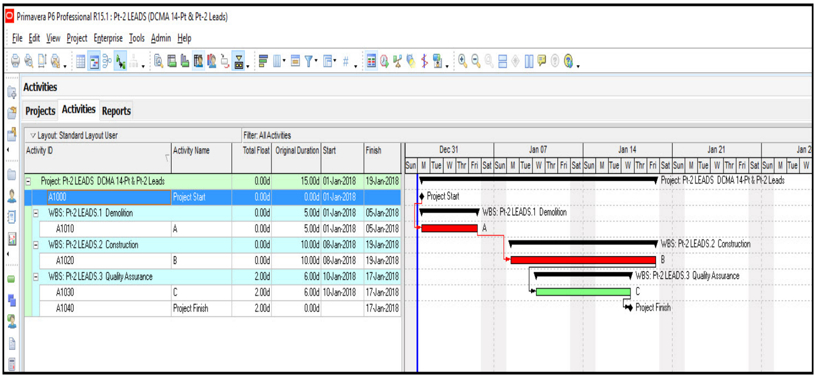


Figure 4

In Figure 4, the lead spread between B and C activities is so wide that the successor activity C actually finishes before the predecessor activity B. This is a clear violation of network logic. Why define a FS relationship between activities B and C, if you let the schedule dishonor it?

Note that activity C and activity ‘project finish’ both have a total float of 2-days, which means they can both be delayed 2-days without delaying the entire project. Well, ‘project finish’ is the end of the project, so these two activities, C and milestone ‘project finish’ should have 0-days total float.

Also note, activity B extends beyond the project finish date, but has 0-days total float when logic would say it should have minus 2-days total float. Note further that activity C is not on the critical path, so we are additionally having issues with our critical path.

What’s happening here is the software says the end of the project is 19-Jan-2018, because activity B extends till that date. Watch what happens when we add a 17-Jan-2018 ‘must finish by’ project constraint, Figure 5.

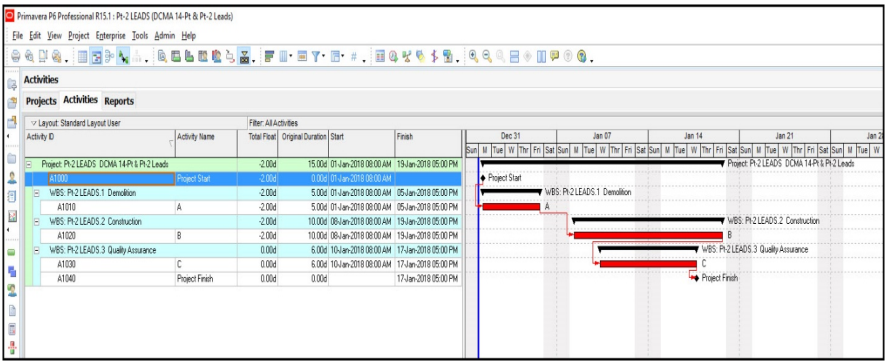


Figure 5

Activity B now correctly has minus 2-days total float. So with the addition of the project constraint we know our schedule has a problem. The fact that activity B has a minus 2-days total float is not good.

Another issue the 14-assessment addresses is the clarity of the schedule. In many cases schedules with leads indicate that the schedule does not contain a sufficient level of detail. Exactly what is the trigger in the FS relationship with negative lag to motivate the commencement of the successor activity before the completion of the predecessor? The narrative of negative lag is better represented by either SS relationships customized with a positive lag or FS relationships connecting shorter duration tasks and no lags.

Let’s demonstrate a schedule that has FS relationships of shorter duration and no leads or lags. First let’s view a construction project that includes lead, Figure 6.

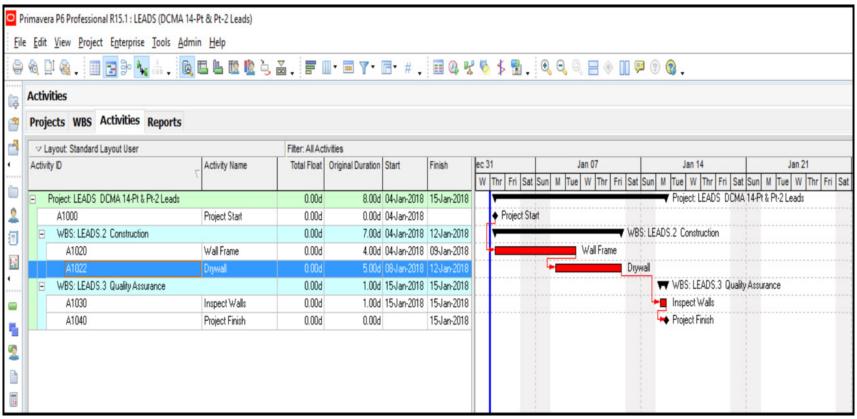


Figure 6

In this schedule, 2-days before you finish installing the wall frame you may commence installing drywall. The question is what is the trigger that allows commencement of install drywall? As you can see, the lead does not provide enough insight. It does not provide the compelling reason drywall installation can proceed, other than 2-days wall framing remaining. In Figure 7 we have the same project and a more descriptive schedule.

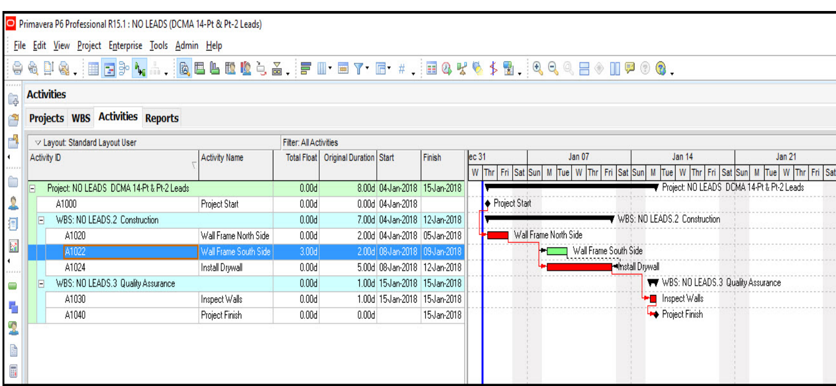


Figure 7

In place of the lead we have shorter (known-scopes-of-work) tasks connected by FS relationships, but no positive or negative lag. Not only does this schedule tell us we can commence drywall after 2-days of wall framing, but we also know the reason. Wall framing the north side of the construction site is complete. After wall framing the north side, we know there is enough framing complete to commence drywall installation. And implementation of the successors Wall Frame South Side and Install Drywall is based on the completion of a known scope of work, Wall Frame North Side. (Note installation of wall framing continues in parallel with the drywall installation.)

The updated schedule tells the construction crew to continue wall framing, but on the south side, and wall framing the south side must finish before we finish drywall installation. The second schedule, having shorter duration well-defined tasks connected by FS relationships and no lead, provides the analyst reviewing the schedule with more and sufficient detail to understand the true narrative of the schedule.

Summary

Despite their support of schedule compression efforts and possibly a nice fit with the schedule narrative, leads are nevertheless forbidden by the DCMA 14-point assessment.

The problem with leads is that they disrupt the forward flow of the schedule at best, and at worst may violate network logic. Leads also indicate that, perhaps, the schedule does not provide sufficient detail. A better alternative is shorter well-defined tasks and FS relationships but no lag, where the trigger for the succeeding activities is a clearly defined known scope of work. This FS relationships and no lags modeling solution is also better than those employing positive lags, which are discussed in the next chapter.

Chapter 4

SCHEDULE 'LAG' INSPECTION

Assessment 1 - Lag Overview

As mentioned previously negative lags are not allowed. Positive lags, waiting time between activities, are allowed, but discouraged. The number of relationships in the schedule with positive lag should be 5% or less. Consider the following formula:

$$\% \textbf{\textit{Positive Lags}} = \frac{\# \textbf{\textit{ of logic links with positive lags}}}{\# \textbf{\textit{ of logic links}}} \textbf{\textit{x}}\mathbf{100}$$

Assessment 1 - Lag In-Depth

One quality of good schedules is that they are clear and understandable. This is at the heart of the DCMA assessment guidelines concerning the use of schedule lags. Let's take a look at this in more detail.

Lag modifies the relationship between two activities by inserting waiting time or delays. Lag helps to better model the true relationship or narrative between activities. The DCMA lag assessment looks at the possible unfavorable impact lag may have on the clarity, accuracy, and dynamic nature of a schedule, and provides guidelines to ensure a quality schedule. This chapter looks at the DCMA (positive) lag assessment as a guage of the quality of a schedule. This chapter does not address negative lag (or lead), which is addressed in the previous chapter on Leads.

The precedence diagram, from chapter two, is displayed in Figure 1, as the starting point for our discussion on positive lags. Recall it describes the cause-and-effect dependencies between two tasks.

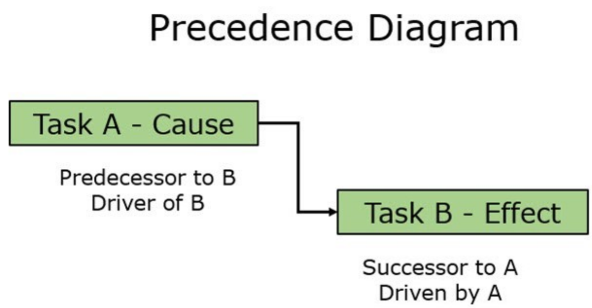


Figure 1

Again, a dependency is the relationship between the (start or) finish of one task and the start (or finish) of another task. As explained in chapter 3 the precedence diagram method supports modifying the dependency or relationship between two activities by inserting waiting time or delays. This waiting time modifier is referred to as lag. Figure 2 displays the precedence diagram of a FS relationship, the most common relationship type, customized with a lag modifier.

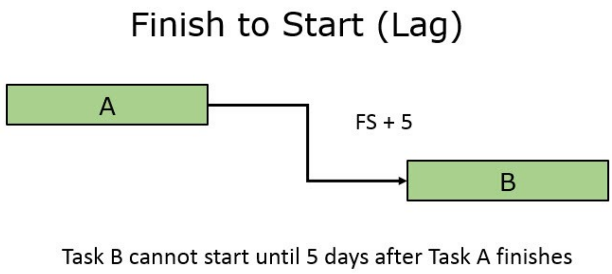


Figure 2

Recall that the lag modifier in this FS relationship means that not only must predecessor activity A be complete before you can commence successor activity B, but you must also wait for a specific period of time after A finishes before you can actually begin the following activity B.

This lag modifier tool helps to better model the true narrative of the connected activities. The classic example of lag is the lag (waiting time) needed to allow for the curing of concrete. After concrete is poured, the concrete must cure before proceeding with a successor task, perhaps, constructing wall framing on a concrete floor. This curing of concrete time is conveniently modeled using lag.

The question at hand now is what do the DCMA 'lag' assessment guidelines have to say? The DCMA 14-point schedule assessment guidelines specifies that to avoid adverse effects on the schedule, and, possibly the critical path, no more than 5% of relationships in the respective schedule should have positive lag. So a limited use of positive lag is acceptable.

A setback in using lag to model schedules is the increased risk due to its lack of visibility. Lag on the Gantt chart is just a line connecting two tasks, and difficult to spot. Lag on both the Gantt chart and activities table lacks visibility and explanation (definition). It is hard to document lag.

A non-apparent or obscure definition of lag in your schedule may have a negative effect, particularly if it is along the critical path. You want the critical path to be clearly defined. But lag is vague and not well-defined or documented, which becomes a problem when trying to detail the true schedule narrative. Lag is therefore not easy to monitor.

So how do you improve your schedule to limit or avoid the use of lag? Well, if the lag represents some outside effort or activity, consider representing it as an explicit task. Indeed the PMI's Practice Standards for Scheduling second edition suggests this is a good alternative to using lag.

Yes, the lag may represent static waiting time for your project team, but, perhaps, your customer might be performing an explicit task. In Figure 3, an original schedule models with an 8-day lag.

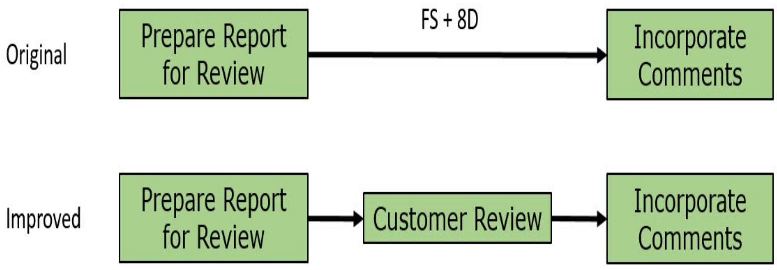


Figure 3

You write and submit a report for review, wait 8-days, and then you incorporate customer comments. During your lag time, the customer is actually performing the explicit task of reviewing the report. An improved schedule, again, Figure 3, would model the lag as an explicit 'customer review' task with an 8-day original duration, and no resource assignments. (Because this effort is beyond your project team no resources and cost loading are associated with this task.) This provides a better understanding to stakeholders of what is actually happening during this 8-day period of time connecting the two tasks. You also can monitor and adjust the customer review activity according to your customer's progress.

It is also better to replace positive lags modeling concrete cure with something more descriptive and transparent. Here no one is performing an explicit task, but a chemical reaction is taking place. The curing of the concrete is better modeled as a task having an estimated original duration, according to the estimated time the poured concrete takes to cure. (Note that a concrete curing task would have no associated resource or costs). DCMA discourages the use of activities without resources, but prefers them to non-descript lags for the same.

Let's take a look at another situation. In Figure 4, we have a simplified precedence diagram of a tile manufacturing project.

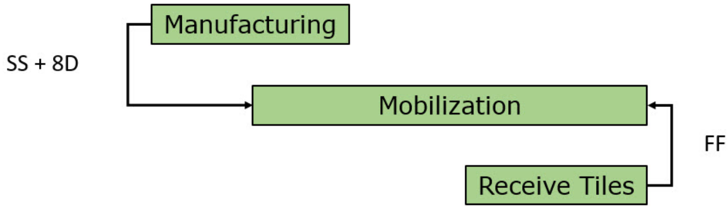


Figure 4

A more detailed Gantt chart screen shot of the equivalent Primavera P6 Professional schedule is displayed in Figure 5.

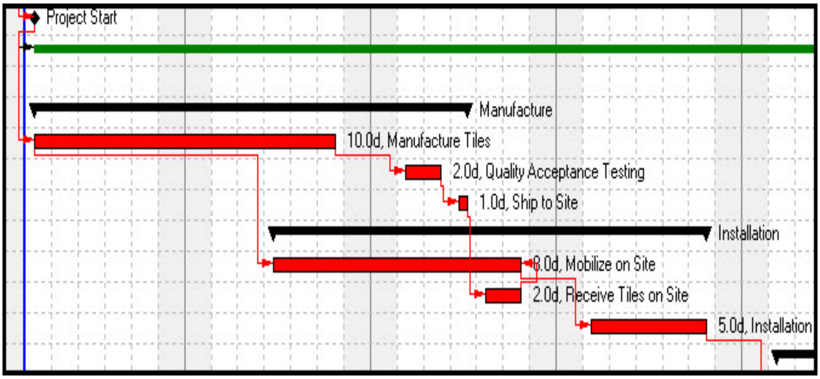


Figure 5

This schedule reads 8-days after we start manufacturing we begin mobilization, and mobilization finishes when all tiles are received on sight. The problem inherent in this schedule is that the 8-day lag appears to be an arbitrary delay. The start of mobilization is not based upon the completion of a known scope of work. What happens if not enough tiles are manufactured to warrant start of mobilization? Stakeholders are unaware of this potential problem, because the lag delay does not inform them of the reason for the delay.

A better solution, Figure 6, breaks up tile manufacturing into two known scopes of work: manufacturing sail tiles and manufacturing main hull tiles.

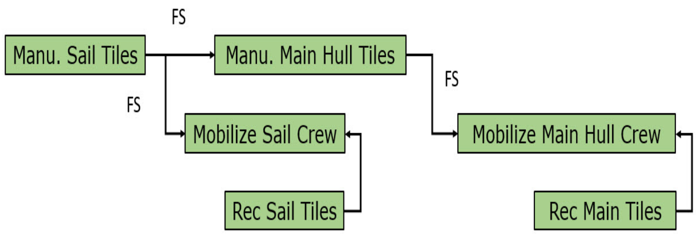


Figure 6

Again, a more detailed screen shot of the equivalent Primavera P6 Professional schedule is displayed in Figure 7. Note, in particular, the P6 detailed screen shot displays the predecessor to Receive Sail Tiles on Site.

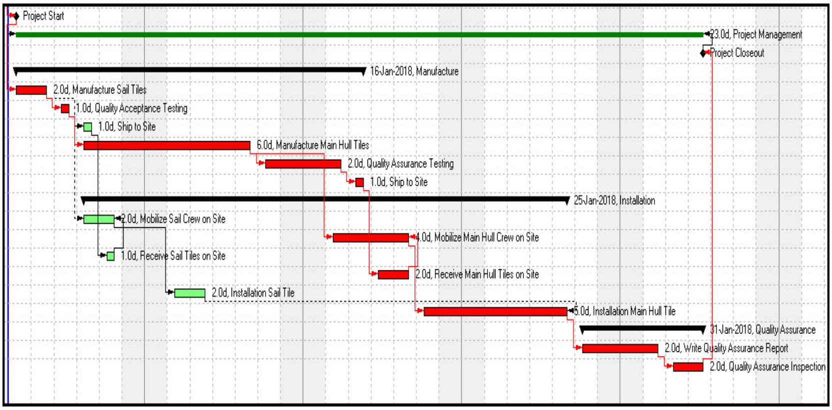


Figure 7

This defined-scope-of-work schedule provides stakeholders a much clearer picture of the project. Taking a step back and inspecting the schedule, they are manufacturing tiles for a scale model submarine. This project has a smaller effort to manufacture and install sail tiles and a larger effort to manufacture and install main hull tiles.

The driver for the start of mobilize sail crew is the completion of the submarine scale model sail tiles manufacturing effort. So instead of the 8-day lag we have a manufacture sail tiles task with 8-day duration. If after 8-days the manufacture of sail tiles is not complete, we know to delay mobilize sail crew until completion of this sail tile manufacturing effort. Thus, defining a known scope of work instead of a lag wait time makes the schedule more transparent.

Another problem with lag is that it is static and simply denotes the passage of time. Lag cannot be updated with progress. Like constraints, lag will not change when predecessor activities are updated with progress, including delays. The unchanging lag may no longer represent the true story of the schedule.

An example is when the general contractor specifies an August 8th start date for an install drywall activity and the schedule logic has the predecessor activity, install framing, finishing on August 3rd, several days earlier, Figure 8.

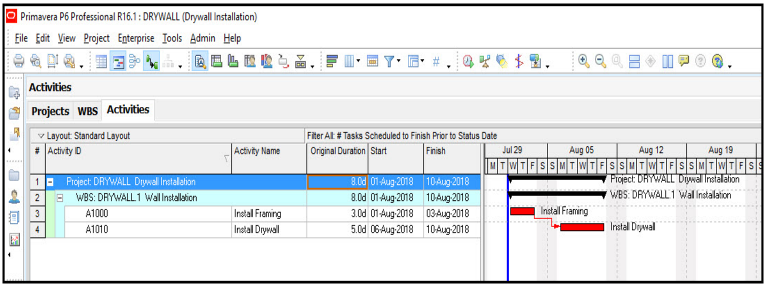


Figure 8

According to your network logic, install drywall would begin on August 6th. Well, to target your drywall installation to commence on the August 8th contractor specified start date you may be tempted to insert a 2-day lag between predecessor install framing and successor install drywall, Figure 9.

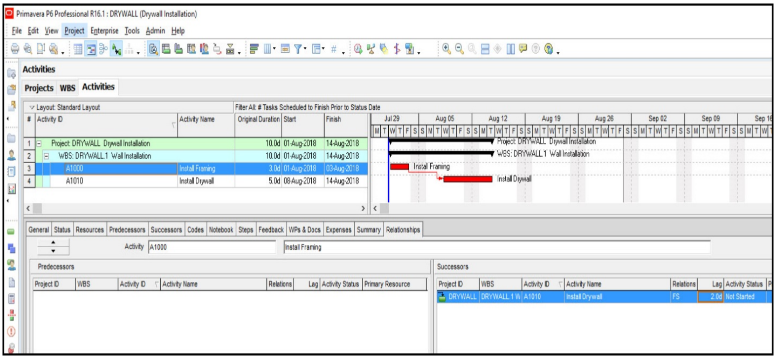


Figure 9

However, if you insert this artificial lag and your install framing activity is delayed, say 2-days, see Figure 10, the 2-day lag time will cause your schedule to overshoot. This will miss the desired August 8th targeted install drywall start date.

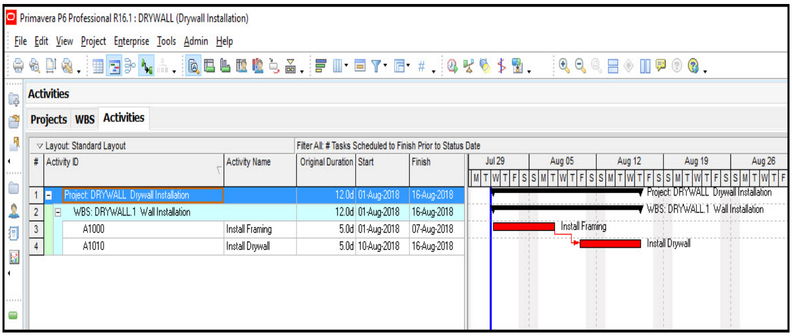


Figure 10

This is because, like constraints, lag does not change with schedule updates (as per our example, the lag time remains 2-days).

Improper use of lag therefore may inhibit the dynamic quality of your schedule. The target date lag must be manually changed as predecessor activity implementation changes. This is another reason that you are recommended to use lag judiciously. Again, limited use of lag ≤ 5% is acceptable. But lag in general is discouraged partly because of its static nature, i.e. its duration remains constant through predecessor activity updates.

Summary

The DCMA negative lags assessment discourages insertion of any negative lags (leads) in project schedules. Positive lags are acceptable, but on a limited basis. In many situations the negative lag is replaced by a positive lag. But positive lags have many of the same disadvantages as negative lags that should discourage their wide adoption.

Just as the quality of a programmer's software code is measured by its documentation and clarity to support other team member programmers, the schedules' true picture needs to be understood by others beyond the actual scheduler. Limit lag in your schedules to no more than 5% of activity relationships to maintain a well-documented and transparent schedule, and reduce risk. Again, unclear lags inhibit the schedule clarity and may adversely affect the critical path definition.

The better solution to positive lag is to define a known scope of work for that particular schedule situation. This is a more apparent and responsive scheduling solution. Consider explicit tasks instead of lag for outside efforts and/or processes, such as the curing of concrete, which may require the insertion of non-resource activities.

While this will conflict with DCMA requirements for non-resource activities, it better describes schedule needs. The lag time may be static nonworking time for your project team, but still represent an explicit dynamic effort or process for others. Finally, note that lag also is static, and, therefore, is not well suited to targeting successor start dates.

Chapter 5

SCHEDULE RELATIONSHIPS

Assessment 4 – Relationships Overview

FS relationships, where the successor can start once the predecessor is complete, are preferred. Use of the SF relationship should be rare and explained in detail. SS and FF are acceptable. Again, FS logic relationships are preferred, and should comprise at least 90% of the schedule's relationship types. The following formula may help inspect the schedule once your schedule is in progress. This analysis is performed only for those activities that are either in progress or haven't been started):

$$\%\ \boldsymbol{FS\ Relationships} = \frac{\#\ \boldsymbol{of\ logic\ links\ with\ FS\ Relationships}}{\#\ \boldsymbol{of\ logic\ links}} \boldsymbol{x100}$$

Assessment 4 – Relationships In-Depth

Of particular importance in the 14-points assessment are the relationships between activities. Activity relationships make schedules dynamic, so that the effects of schedule change propagate through the whole schedule. This means that activity start and finish dates automatically respond and update to changes in the schedule or to progression of the schedule. So it's important for schedules to be responsive to input. But it is not enough for a schedule to be dynamic, it also must be understood by stakeholders. Much of the clarity of a schedule is determined by the types of relationships defined between activities. Transparent schedule relationships are therefore an important measure of a schedule's quality. For this reason the DCMA 14-point assessment prefers the FS relationship for defining the interface between activities. In this chapter we take a closer look at why the DCMA assessment prefers the insertion of FS relationships to define the interface between activities.

The FS relationship is the most common scheduling relationship. The precedence diagram for a FS relationship, Figure 1, says Task B cannot start until Task A finishes. The tasks in the FS relationship are sequential, activity B takes place after activity A.

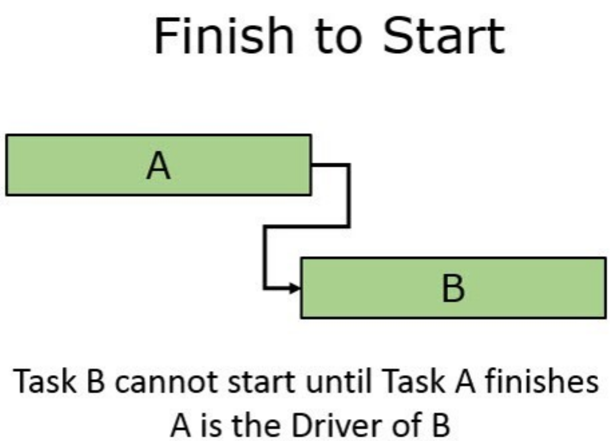


Figure 1

Recall from Chapter 2 that in addition to the FS relationship schedulers also have SS, SF, and FF relationships in their arsenal of relationship tools. Precedence diagrams for these relationships are again displayed in Figures 2, 3, and 4. Note, as discussed earlier in Chapter 2, that the SF relationship precedence diagram, Figure 4, includes lag, because this presents the most practical application of the SF relationship.

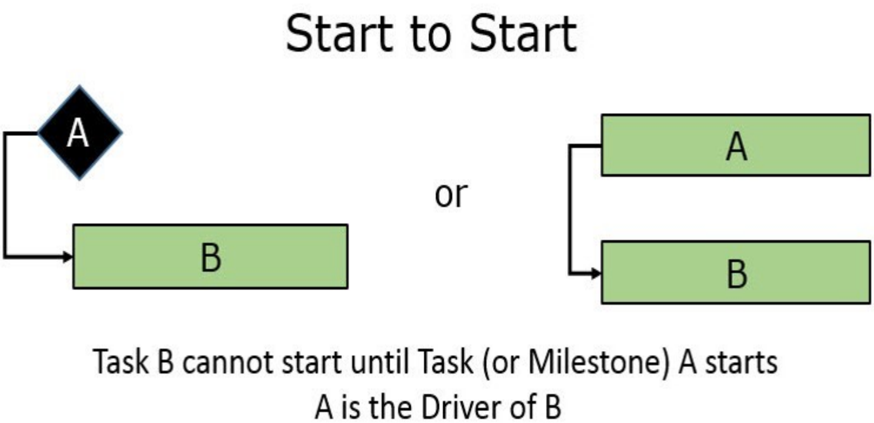


Figure 2

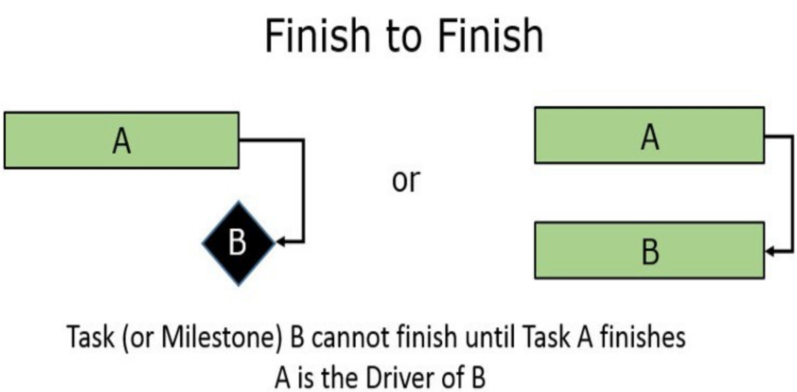


Figure 3

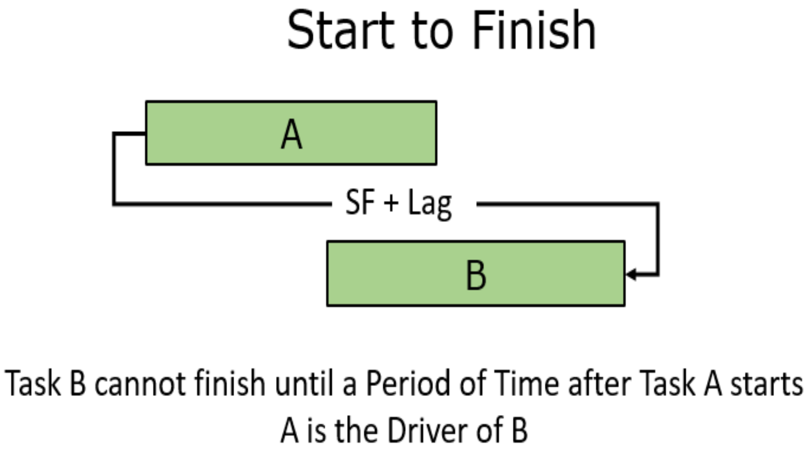


Figure 4

All four of these relationship types are acceptable according to the 14-point assessment. Note, however, that the SF relationship type is difficult to understand. Without delving into further detail about the SF relationship, it is best to limit the insertion of SF relationships in the schedule to extremely rare situations.

The 14-point assessment does not forbid SF relationships nor any of the other relationship type. It does, however, specify in the interest of schedule transparency that at least 90% of all activity relationships should be of the FS type. So no more than 10% of all schedule task relationships may be SS and FF types.

Though their usage should be limited, SS and FF relationships, in particular, are acceptable in situations where they describe the true nature of the dependency.

SS dependencies

1. Demolition cannot proceed until commencement of the safety plan.
2. Level concrete cannot proceed until commencement of pour concrete.
3. Monitor orbit cannot begin unit start of launch rocket.

FF dependencies

1. Write report cannot finish until research is complete.
2. Report editing cannot complete until report writing efforts finish.
3. Student evaluations cannot complete until the class is done.

If these activities, however, do not truly have the respective SS and/or FF dependency; SS and FF relationships should not be used to simply fast track a schedule, (i.e. shorten a schedule by taking sequential activities and performing them in parallel).

As the FS relationship is sequential, where activity A must complete before progressing to activity B, it is easy to understand. And, as mentioned above, it is partly for this clarity that the 14-point assessment encourages the use of the FS relationship. Another reason FS relationships are preferred is they support defining known-scopes-of-work tasks in the schedule. Recall from chapter 4 – lags assessment that both positive and negative lags have visibility problems; they both appear as simple lines on the Gantt chart and do not document or make apparent the true schedule situation. Instead of inserting a SS or FF modified by a positive lag define known-scopes-of-work tasks connected by FS relationships, and no lags.

Shorter well-defined tasks that have FS relationships are preferred. Refer to the chapter 4 scale model submarine tile manufacturing project for a good example of removing positive lags and instead inserting shorter duration known-scopes-of-work tasks connected by FS relationships. Making this lag replacement may require viewing the schedule from a new vantage point to see how your schedule lag is replaceable with shorter known-scopes-of-work tasks and FS relationships.

Summary

So be extremely cautious and question the project team as to the requirement for the need to insert an SF relationship. And SS and FF relationships are appropriate where a true dependency is present.

However, for schedule simplicity and transparency use of FS relationships is encouraged. Other relationship types are more difficult to trace, justify and can introduce excessive float. So instead of SS and/or FF with lags connect known scopes of work with FS relationships. As a general guideline, 90% of schedule dependencies should be FS.

Chapter 6

HARD CONSTRAINTS

Assessment 5 – Hard Constraints Overview

Constraints are additional stipulations on dates to further define the importance of those dates in the life of the project. Constraints are soft or hard. Soft constraints may generate negative total float, but will never violate network logic. Hard constraints, however, may fix dates in the schedule in such a way that dishonors network logic dependency relationships. Hard constraints are not forbidden, but they are discouraged. No more than 5% of tasks in a schedule should have hard constraints. Inspect the schedule using the following hard constraints formula:

$$\% \, \textit{Hard Constraints} = \frac{\textit{Total \# of Incomplete Tasks with Hard Constraints}}{\textit{Total \# of Incomplete tasks}} x100$$

Assessment 5 – Hard Constraints In-Depth

Hard constraints are date constraints on schedule activities that may cause them to violate activity dependency relationships, i.e. the network logic. DCMA 14-point guidelines say they should encompass no more than 5% of incomplete activities in the schedule. This actually is quite generous (or lenient) for the DCMA 14-point assessment. I thought they should be completely discouraged, in much the same way that negative lags (leads) are discouraged by the guidelines. Still, as mentioned, limited application of hard constraints is acceptable according to the DCMA 14-point assessment. This chapter discusses the DCMA hard constraints assessment as a monitor for the quality of a schedule.

Schedule activities describe the work required to produce the deliverables, which are the whole purpose for the project. These deliverable producing activities or tasks have relationships that define the interaction between them. Tasks may also have constraints that provide further definition and restraint. These constraints are date restrictions placed on activities because of important influences affecting execution of the project.

Constraints come from contractual agreements, external influences, and/or internal considerations. A contractual agreement may include a delivery date or other deadline in the contract. Missing a contractual date may result in some financial or other penalty. External constraints may include winter shutdown periods, e.g. a dredging company must be out of the water by November 15^{th}, when the river freezes. Environmental regulations are common external constraints. An example environmental constraint includes restrictions on construction work near a river during fish spawning season.

Internal constraints are restraints that the project manager may have some control over. An example includes a resource with scheduled vacation during a particular activity effort. The project manager may replace this over allocated resource with another suitable resource with no conflicting vacation plans. In this situation, again, the resource availability task date constraints are avoided by switching out the resource that has conflicting vacation plans.

Let’s pause our constraint discussion to review network logic relationships from chapter two. Activity relationships, i.e. the cause-and-effect dependencies between activities, are defined using the precedence diagram, which is displayed in Figure 1.

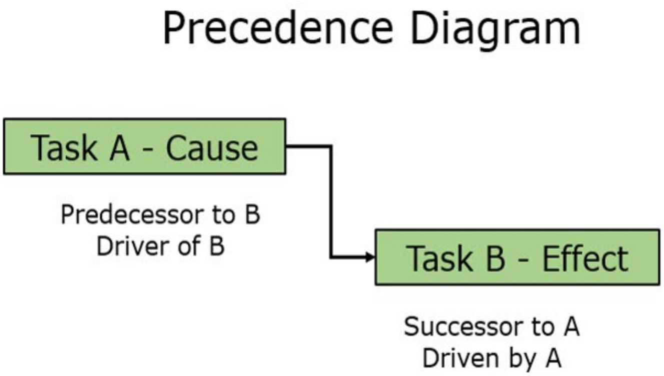


Figure 1

Note the line connecting the two tasks defines the relationship or interaction between them. Figure 2 displays the most common dependency, FS, where the predecessor cause (task A) must be completely done before commencing the successor effect (task B).

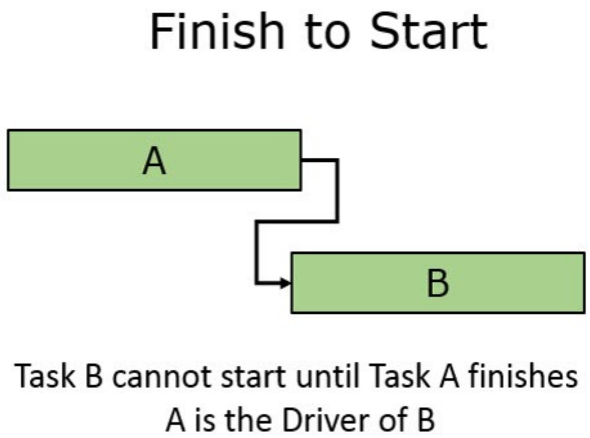


Figure 2

Schedules that have these or similar driving dependencies, perhaps, SS or FF, are called dynamic, as any changes in durations or any progress updates automatically propagate through the schedule. Thus, activity dates for the entire schedule are automatically updated, accordingly, when you have a dynamic schedule.

Constraints, as mentioned above, are date restrictions placed on tasks and relationships as described above. The intent of constraints is to provide further task definition to better describe the true narrative of the schedule. The main issue with constraints is that they make your schedule static, i.e. unable to automatically propagate changes and updates. Hard constraints are unique and problematic in that the constraint date takes precedence over any associated task relationship. Many times this means that the task dependency relationship is violated in order to honor the constraint date.

Let’s investigate. In Figure 3 we have a Microsoft Project schedule with FS activity relationships and as soon as possible (ASAP) soft constraints.

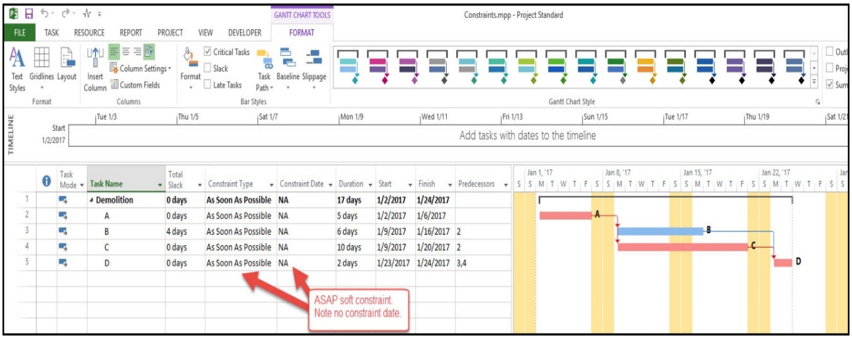


Figure 3

ASAP and as late as possible (ALAP) constraint types are both considered soft, and have no constraint date assignments. ASAP and ALAP constraints simply shift an activity, accordingly, within the activities available total slack. Total slack or total float (in days) is the period of time that an activity can postpone without delaying a constraint date or the entire schedule. Activities that have a total float of zero cannot be delayed at all or the project comes in late. Negative values of total float on activities mean that these tasks are already late, and causing project schedule delays.

You know you have a problem, if any of your activities are displaying negative total float. Note that ASAP and ALAP soft constraints will not generate negative total float. Other soft constraints, however, may result in a negative total float value. This may also depend on your scheduling software.

In Figure 4 we insert a start no earlier than (SNET) soft constraint in our Microsoft Project schedule.

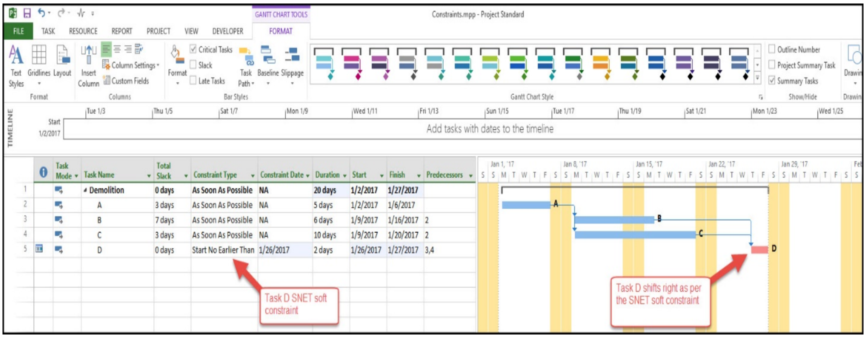


Figure 4

Note that task D is shifted right to start on the constraint date. Great! So activity D was shifted right to start on the constraint date, and network logic was not violated. This is what we want. Figure 5 is the same project, but the soft constraint is replaced with a start no later than (SNLT) hard constraint.

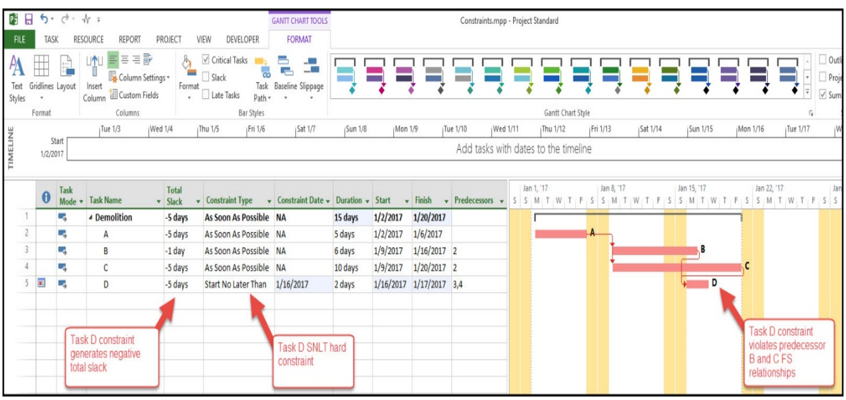


Figure 5

The SNLT constraint date is January 16, 2017. The schedule meets this hard constraint date, but in doing so it had to violate network logic, i.e. dependency relationships. Note on the Gantt chart that activity D starts before the completion of both tasks B and C, which is a violation of the FS relationship between task D and these predecessor tasks.

So in order to honor the hard constraint date on activity D the schedule violated the FS relationships with task D's predecessors. The one saving grace in Microsoft Project is that the total slack column in the task table displays negative values, which is your clue that something with the schedule is not quite right.

Hard constraints come in different flavors depending on your schedule software, most notably Microsoft Project and Primavera P6. In Microsoft Project there are four hard constraints that may violate schedule logic: Must Start On (MSO), Must Finish On (MFO), SNLT, and Finish No Later Than (FNLT). All these constraints may violate the predecessor/successor relationship in the precedence diagram. Depending on the constraint date, all these constraints may display the successor on the Gantt chart occurring before the start or completion of the predecessor. This does not make sense, in particular, if you have a FS relationship between the two activities. In the FS relationship, the predecessor must completely finish before you can commence the successor. The above listed Microsoft Project hard constraints may violate this FS relationship, so the FS relationship successor actually comes before the completion of the predecessor.

In Primavera P6 Professional there are only two constraints that may violate schedule logic: mandatory start (MS) and mandatory finish (MF). The start on (SO), finish on (FO), start on or before (SOOB), and the finish on or before (FOOB) constraints will not violate network logic, but may display negative values for total float in the activities table. Again, whether you are dealing with a soft or hard constraint, negative total float values are your clue that your schedule is falling behind.

Hard constraints breed confusion because they may indicate on the Gantt chart that the project achieves the planned completion date but, possibly, only by violating Gantt chart network logic. So the hard constraint's completion date may be fictitious. In reality, when the Gantt chart network logic is honored, the schedule may not meet the planned completion date specified by the mandatory finish hard constraint.

This is why hard constraints should be limited to no more than 5% of the total number of incomplete tasks, according to the DCMA hard constraints assessment. So, hard constraints should be few in number and should include a note of explanation. The DCMA counts on your knowledge of total float to understand that negative total float is your warning that your constraint is causing the schedule to go amiss.

It is possible in Microsoft Project to un-toggle the schedule option 'tasks will always honor their constraint dates', as per Figure 6.

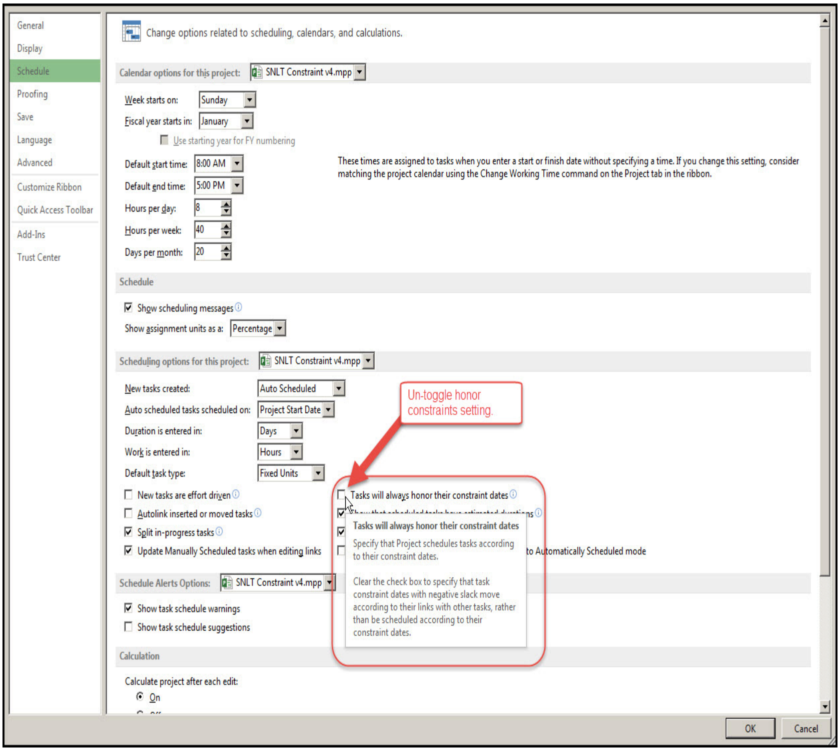


Figure 6

If you un-toggle this setting then Microsoft Project will not breach and disregard task relationships on the Gantt chart, and will still display negative values for total slack

in the task table. Figure 7 is our SNLT constrained schedule with this toggle setting off.

Figure 7

In Figure 7 there are no Gantt chart network logic violations, and we note the negative total slack and additional warning that the task starts after the constraint date. Toggling off 'tasks will always honor their constraint dates' appears to be our solution. But in reference to the DCMA 14-point assessment, we assume that this toggle is on when the assessment reviews hard constraints. So Microsoft Project hard constraints, as well as Primavera P6 hard constraints, remain an issue for DCMA.

Summary

Constraints, in addition to relationships, help define the true story of the schedule. Constraints are either soft or hard. Soft constraints will not infringe network logic. Hard constraints, however, may violate task dependencies. This is the main reason that hard constraints in the DCMA assessment are limited to 5% of uncompleted activities.

You want your schedule to have an unbroken flow of logic from start to end. Hard constraints interrupt and break that logic and cause confusion, particularly, on the Gantt chart. DCMA understands this possible unfavorable impact of hard constraints, and limits their use accordingly. The DCMA hard constraints assessment, however, is not judicious or cautious enough though. Constraints of any type, both soft and hard, should be limited as they render a schedule static, i.e. inflexible to changes and updates.

Yes, the DCMA hard constraints assessment helps determine the quality of schedule, but it does not go far enough in ensuring both a logical and dynamic schedule. Consider the DCMA hard constraints assessment as a bare minimum quality control restraint on the use of activity constraints. A routine review of schedule constraints should be a part of each schedule update cycle to assure that the bare minimum constraints necessary have been defined in the IMS

Chapter 7

HIGH FLOAT

Assessment 5 – High Float Overview

Tasks that have total float greater than 44 working days (2-months) are captured in this metric. High float intuitively is thought to be a good thing; it provides your schedule a safety margin. But excessive high float may be indicative of missing relationships. The number of high float tasks should not penetrate the 5% allowable threshold. Inspect the schedule for high float using the following formula:

$$\% \, \textbf{\textit{High Float}} = \frac{\textbf{\textit{Total \# of Incomplete Tasks with High Float}}}{\textbf{\textit{Total \# of Incomplete tasks}}} \textbf{\textit{x}}\textbf{100}$$

Schedules that have a high float greater than 5% are flagged for review of potentially unstable or missing network logic.

Assessment 5 – High Float In-Depth

The intention of the high float assessment is to avoid activities that are missing relationships and/or have weak relationships. High float activities may be indicative of missing predecessor and/or successor relationships. Schedules that have many high float activities are more susceptible to these missing relationships. A high float activity in your schedule may actually be a “dangling” activity that does not have a proper successor. This requires inspection of the activity successor relationship.

The DCMA high float assessment limits high float activities to 5% because of the propensity of high float activities to behave as “dangling” activities that are not properly tied into the schedule. Each high float activity in the schedule requires inspection to verify that it is not missing logic or, perhaps, needs better logic. This chapter examines the DCMA high float assessment as a means to flag potentially unstable network logic.

DCMA says the maximum acceptable task total float is 44-days. Well, why one single value? Why doesn’t the maximum acceptable task total float duration adjust or scale with the size of the project? It appears to be one size fits or doesn’t fit all schedules. Well, the illustration I like to use is a forest. Let’s say you are in a helicopter flying above the forest. Regardless of the size of the forest, vast or small, you have to be at about the same height or altitude above to *distinguish the trees from the forest*. In schedule analysis, regardless of the total duration of your project, you want to be at the appropriate height or scale for viewing and understanding tasks and their associated predecessors and successors.

One may still ask: why is 44-days total float the derived cutoff point for achieving this perspective? Well, there appears to be nothing magical about the duration 44-days except that is comes to about two months, or roughly two reporting periods in a typical earned value management integrated system. Two months total float duration is a value, most likely, derived from practical experience and/or analysis on schedules of varying length. We are taking DCMA’s 44-day high float definition on face value. Again, a maximum 44-days total float appears to provide about the right zoom or panorama for analyzing the schedule and reporting progress. Thus, the restrictions on activities having greater than 44-days total float.

But shouldn't positive total float always be considered good? The total float of a task is the number of days it can delay without delaying a constraint date or the end of the project. Yes, and project managers prefer tasks that have positive total float as it means they may delay the activity without negatively impacting the schedule. Again, positive total float is like a built in safety margin. If a task does not progress according to plan it still might be acceptable on the schedule's time scale, as per the remaining value of its total float. So positive total float is good.

But why are high float tasks unfavorable or suspect? Again, a high float task may indicate missing predecessors and/or successors, and that the schedule network may not be logic-driven. In light of these high float possible negatives we recommend schedulers filter all tasks according to total float, and flag all high float tasks for review. Inspect the task logic to ensure that predecessor and/or successor relationships are properly defined, and do not leave the activity "dangling".

A good example of a high float task missing a successor is an install perimeter fence task on a construction site. On this project the install fence activity has, let's say 60-days total float. This means the install fence activity can delay 60-days and still not negatively impact the conclusion of the project.

In our example, note that the total float of install fence extends to the end of construction, which, realistically, it can. But this may not be our desire. Most likely, we want our contractor to install the fence earlier because it provides security to the job site. Thus, our dilemma: we have lots of total float, which is great, but we prefer less because we want to secure the jobsite sooner.

One solution is we could put a finish on or before constraint on install fence to get it built earlier. However, this may seem like an arbitrary date, particularly, if schedule progress does not go as planned. And it also may appear less than honest, as no good reason to commence the project at the constraint date is provided; the constraint may simply be a means to hide total float from the subcontractor to keep them from delaying construction. And a constraint may not truly represent our schedule situation.

A better solution, more true to our project situation, is to complete the fence installation just prior to delivery of the bus and jumper supports, high voltage dangerous equipment. The thought is you want to restrict access to the construction site before delivery of this dangerous equipment. So the perimeter fence serves its purpose as job site security long before the completion of construction.

Let’s investigate install perimeter fence in light of this more accurate narrative. Analysis of our high float fence construction task reveals it is missing its FS link to the true successor receive bus and jumpers. Completion of install fence should precede delivery of bus and jumpers. Thus, we have a situation where a task has high float because it really is missing a successor. And it is not properly depicting the true portrait of the schedule. In this case the schedule is not driven by the logic of securing the construction site perimeter on or before delivery of the bus and jumpers.

Summary

Provided that a critical path has been established, positive total float is good. It's what project managers are hoping for. It provides a "cushion" for unpredictable delays. Particularly high float tasks, however, can be prognostic of missing predecessors and/or successors. The DCMA 14-point assessment proposes that tasks that have total float greater than 44-days (2-months) are suspect, and should be examined closely.

The DCMA 14-point assessment does not forbid high float activities; it limits their use to no more than 5% of incomplete tasks. They are limited even in logic-driven schedules, perhaps, for no other reason than that they make it more difficult to track and/or understand the true story of the schedule. Therefore, we recommend that the schedule team routinely identify all tasks with suspicious high float of =/> 44 days. You should verify that all relationships have been identified with the associated activities to ensure that float values are minimized.

Chapter 8

NEGATIVE FLOAT

Assessment 7 – Negative Float Overview

Total float is a measure of how long an activity may delay without delaying the entire project. Positive and zero total float are acceptable. Negative float, however, means your schedule is already behind, which is not good if you're still at the planning stage. The negative total float metric captures tasks that have total float less than zero. Tasks with negative float should come with a transparent note of explanation and a negative total float mitigation plan. Use the following formula to compute negative float percent.

$$\%\,\boldsymbol{Negative\ Float} = \frac{\boldsymbol{Total\ \#\ of\ Incomplete\ Tasks\ with\ Negative\ Float}}{\boldsymbol{Total\ \#\ of\ Incomplete\ Tasks}}\boldsymbol{x100}$$

This formula is simply a guage of how bad the schedule float situation is, because realistically there should not be any negative float in the schedule.

Assessment 7 – Negative Float In-Depth

The negative float assessment notes the number of incomplete tasks with negative float. The negative float assessment fails if any task has total float less than zero. Tasks that have negative float, again, should have an "explanation and a corrective action plan". This chapter examines the DCMA negative float assessment as support for negative float monitoring and respective adjustments to eliminate any vestiges of negative total float in the schedule.

Total float, also referred to as total slack, is computed from the forward and backward network passes and its value is the Late Dates minus the Early Dates. Total float is a property assigned to each task or milestone in the schedule. Total float values can be positive, zero, or negative numbers in days.

A positive total float task may slip according to the value of its positive total float. Yes, positive total float is good. It's what project managers are hoping for. It is like a built in safety margin. It means you may delay an activity, accordingly, and still not affect an activity constraint date or the project completion date.

Zero total float tasks require attention. When a task has a zero total float value, the task cannot slip at all. Any delay on a zero total float task means you will miss either a target activity constraint date or the project completion date. Not good! But the longest path through the schedule network are all connected tasks that have zero total float. So zero total float activities are not necessarily bad; they tell you the duration of the project schedule. However, as noted, zero total float tasks require attention. They do not have a delay safety margin or buffer.

Negative total float tasks are your warning something is amiss in the schedule. Negative total float tasks indicate probable failure to meet one or more schedule target completion goals. Negative total float means you are in danger of delaying an activity constraint date or the entire project or activity relationships are either missing or connected incorrectly. And a negative total float value is generated on every upstream task that is linked in the potential target date failure. This is good, because it tells you which activities require optimization adjustments to realign the schedule in keeping with target dates.

Okay, you have monitored your schedule and found negative total float activities. How do you mitigate this negative float? Well, again, your negative total float tasks are warning you that your schedule duration in light of desired completion dates is too long. Implementation of the schedule as is will result in missing these target dates. There are a number of optimization efforts schedulers can implement, which will shorten the schedule and remove negative float.

Create Parallel Paths From Sequential Paths (Fast Tracking)

Many times the biggest bang for the buck comes from changing activities in series to activities in parallel. So when optimizing the schedule look for sequential relationships that are not necessary. Many times schedulers will include two tasks in a FS sequential relationship because they assume that the successor activity is performed by the same crew working on the predecessor activity. This may in fact not be so, in which case you may have a soft dependency at most. And you can schedule the two activities concurrently, so they are performed in parallel.

Performing soft dependency activities in parallel instead of sequentially can be a huge time saver. This is known as fast tracking the schedule. A simple example, Figure 1, is that install windows and install doors do not need to be done sequentially, performing them in a parallel SS relationship makes sense, Figure 2.

Figure 1

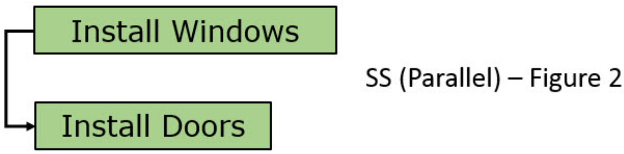


Figure 2

Particularly, if there are, again, separate resource crews working on the two tasks and there are no issues with resource crews competing for work space.

Here, quality and scope will not be impacted, but the possible time reduction is significant. It's important to note that DCMA will question the use of SS relationships. So, you must have sufficient Quantitative Backup Documentation (QBD) to substantiate your actions.

Change Sequential Dependencies Into Partial Dependences (Fast Tracking)

Changing a sequential dependency into a partial dependency is only possible if you do not have a hard ("set in stone") FS relationship between the two activities. In a hard FS relationship you are saying that the successor activity cannot start until the full completion of a predecessor activity. A classic example is when your predecessor activity is lay foundation and your successor activity is install wall frames, Figure 3.

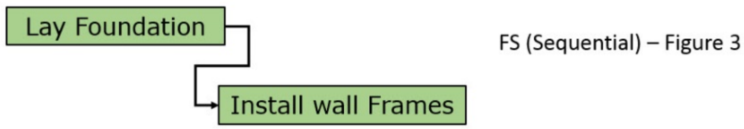


Figure 3

You absolutely must be finished with laying the foundation before you can begin installing walls. But what about the activities dig trench and lay pipe for an underground pipe installation. Dig trench is the obvious predecessor and lay pipe is the successor. Now we know that you must complete dig trench before you can complete lay pipe. But must your dig trench activity be completely done before the commencement of your lay pipe activity? The answer is no. So you can change your sequential dependencies into partial dependencies. This, again, is referred to as fast tracking the schedule.

One possible modeling solution is that you have a SS relationship between the predecessor dig trench and successor lay pipe activities customized with a positive lag on the successor lay pipe activity, Figure 4.

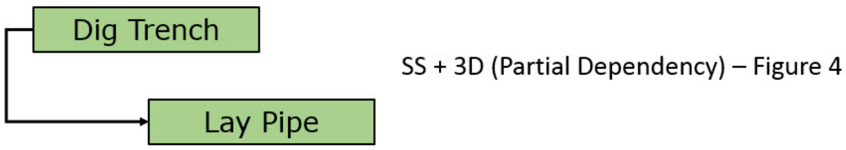


Figure 4

Here the lag is waiting time (3D – three days) between the start of a predecessor activity and the start of the successor. Using the lag allows you to delay laying the pipe activity several days until there is enough room in the trench to proceed with installing the pipe. As before, don't forget to have adequate QBD to substantiate your actions.

Others recommend changing the original FS relationship by adding a negative lag. (Another name for negative lag is lead.) So you adjust your FS relationship by adding a negative lag or lead, Figure 5.

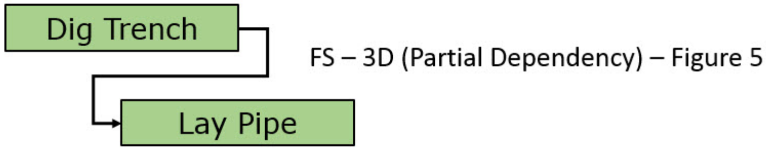


Figure 5

Here the negative lag or lead says you can proceed with the successor activity a period of time, say 3 days, before the completion of the predecessor activity. Negative lags are not recommended and are discussed in Chapter 3 Pt-2 Leads.

If your absolute preference is for the use of negative lags then use them sparingly. And be forewarned that many government agencies forbid the use of negative lags and, therefore, will not approve your schedule if you include them.

It is generally thought that making sequential dependencies partial dependencies will not impact the quality or scope, but will reduce time. Great! This is what you want.

Adding Resources (Crashing)

Adding additional resources to a task, which is known as crashing the schedule, is a way to cut down on the task's time without sacrificing scope. This will most likely increase cost, because the time savings of a shortened activity usually does not result in a comparable resource cost savings to offset the added cost of additional resources. So with crashing your time supposedly goes down, but your cost, most likely, will increase.

Others argue that you might not even get the time savings you hoped for. They say this because your additional resources may not be as skilled, requiring training from your more skilled workers. Therefore, there is an initial ramp up time for new and/or unfamiliar workers to become productive. This results in an initial early slowdown in the hopes of gaining time later on, and that may not happen.

Another important consideration before crashing an activity is work space. Will you have enough work space for all your additional workers to be productive and actually increase crew productivity? If your construction work is on a submarine, the answer may be no.

Also consider carefully what is known as the *law of diminishing returns* when adding resources. This is where adding more and more resources results in less and less impact on the schedule time reduction. There also comes a point where adding more resources may not improve productivity it may actually hinder it. This could be caused by increased communication issues with larger crews or other factors, as working space described above.

Here the time savings is questionable. The effect of using untrained resources makes the quality questionable. The scope, however, should have no impact. Many times a project manager's gut reaction is to add resources to compress the schedule, but this just is not the best first option as described above.

What Else To Look For?

Rather than add resources look for these additional time saving possibilities:

- **Review the estimated durations of critical activities**. Perhaps, the scheduler "padded" the estimated activity duration. Padding is where the estimator adds extra time to an activity because they do not have enough information to make an accurate estimate. Decreasing duration estimates benefits the schedule time, but may impact quality and, perhaps, even scope. Decreasing the duration of any task on the critical path will be questioned by DCMA, so don't forget your QBD.

- **Review the estimated durations of lags on the critical path**. Again, lags are waiting times between activities, such as the time to let a concrete foundation cure before installing the walls on the foundation. Perhaps, you really do not need one week for your concrete to cure before commencing your successor activity. Also, does your concrete cure task calendar account for weekend cure time. The benefit is reduced time, but, perhaps, at the expense of quality. Scope is not affected.

- **Look for extremely long duration critical tasks**. These may be split into smaller tasks and some of these resulting smaller tasks may be optimized by performing them in parallel. Ideally, this should not affect quality and scope, but possibly reduce time.

- **Look at critical path project scope**. Are all the deliverables necessary or can deliverables be removed and, therefore, activities eliminated from the critical path? Of course, this reduces scope, and, therefore quality, but the time savings may be significant and of primary importance. Its tough business deciding to cut off tree branches, but if they are dead weight they have to go. And it just might save the tree and/or project.

Summary

The normal condition of successful projects is zero or positive float tasks. Again, the zero float tasks reveal your critical activities and longest path. Positive float is great; it's like having a built in safety margin, which allows for unpredictable delays. Negative float tasks, however, indicate that your schedule is already behind, and target dates cannot be achieved.

The DCMA negative float assessment, which is run against schedules in the late to completed planning stage, fails any schedule that has negative float tasks. The negative float assessment further specifies that negative float activities should have an explanation and a negative float corrective action plan.

The implication here is that the schedule is approved pending implementation of the negative float mitigation action plan. So consider the many ways to optimize your schedule to limit negative float, and develop your negative float mitigation action plan for review. Remember to have adequate QBD's in anticipation of DCMA asking for justification for your actions.

Chapter 9

HIGH DURATION TASKS

Assessment 8 – High Duration Tasks Overview

A high duration task is any task that has a duration greater than 44 working days (i.e. 2 months). Excessive duration tasks may cloud visibility of schedule status and progress. The percentage of schedule high duration tasks is computed as follows:

$$\% \, \boldsymbol{High\ Duration} = \frac{\boldsymbol{Total\ \#\ of\ Incomplete\ Tasks\ with\ High\ Duration}}{\boldsymbol{Total\ \#\ of\ Incomplete\ tasks}} \boldsymbol{x100}$$

High duration tasks should not exceed the 5% allowable threshold or the schedule fails this assessment. The exceptions include long lead procurement activities or level of effort tasks, such as project management and administration.

Assessment 8 – High Duration Tasks In-Depth

High or long duration tasks have the following disadvantages:

1. They make discrete progress measurement more difficult.
2. They increase the possibility of required changes to in-process work.
3. They may distort the critical path.
4. They are more susceptible to the negative lag relationship modifier.

The DCMA high duration task assessment places limits on the maximum allowable task duration. But what about long duration projects? Should the task durations of lengthy projects be limited to 44-days? To use an extreme example, a former student that works for NASA estimated that a mission to mars; including concept, prototypes, final design, production, launch, and travel to mars, would have a duration of, perhaps, 20 years. If this is your project duration situation then what would you make the limit or maximum duration of a single task? This chapter discusses the DCMA high duration tasks assessment as a measure of the quality of a schedule.

Rolling Wave Planning and Discrete Work Packages

When it is not possible to foresee all future activities and discrete work packages in a project, implement *rolling wave* planning. Project planning in *rolling wave* is done in stages or "*waves*" where activities in the near term are planned in detail and activities farther out in time are left for future planning. *Rolling wave* near term activities are discretely defined but farther out activities are more generally defined and more pliable. These farther out activities should, perhaps, include the labeling 'planning package' in their title name. So 'planning packages' represent far-term effort where definition of discrete work packages is presently unknown. The 'planning packages', most likely, would represent the 5% acceptable long duration task threshold. This way the schedule captures all scope, and near term efforts are discrete work packages. This is common practice in earned value project management.

Long Duration Tasks and Discrete Measurement

Again, what would you make the limit or maximum duration of a single task on a long duration project, say a

20-year project? The answer to this question depends on the level of work package decomposition required to make activities discrete enough to track and manage. Again, the DCMA high duration task assessment says that no more than 5% of tasks should have a duration greater than 44 working days. So 44-days is the cutoff.

Why 44-days you may ask? Is there something magical about this 2-month period of time? Well, tasks greater than this are more difficult to track and manage. Also, schedules are commonly tracked or progressed on a monthly basis. One would expect tasks to either start or finish during these two normal status cycles. So excessive duration tasks (in a seemingly endless in-progress state) make it harder to get visibility into the project schedule status and/or progress. Because of this, both short and long duration projects all have to meet this requirement. Long duration tasks make discrete measurement more difficult. Shorter tasks provide more clarity and accuracy of the actual work to perform. Shorter tasks are more measureable and yield more visibility into the project's discrete performance.

Long Duration Tasks and In-Process Work Changes

Long duration tasks also increase the possibility of required task changes to in-process work. In this situation the task is so long that part way through implementation it is discovered that an adjusted definition of the work is required. Changes in work definition may affect the baseline and require sending the schedule through baseline update control procedures.

Long Duration Tasks and Critical Path Distortion

High duration tasks may also distort the critical path. The thought is that a long duration discretely defined task is not accurate, and, therefore the associated task duration estimate is also incorrect. Some prefer longer tasks with interim milestones depicting discrete performance. But, in reality the same tasks could be broken up into each of the interim milestone work package segments and yield better visibility into its overall performance. Distortion of the critical path either to the long side or short side is the likely result. As defined above, you would be better served by implementing rolling wave planning. This would show short/near term tasks that are discretely defined and far off efforts that are described in high level ‘planning packages’.

Long Duration Tasks and the Negative Lag Modifier

Long duration tasks are also more susceptible to the negative lag relationship modifier, which is another DCMA lead assessment described in chapter 3. Note that leads are discouraged and even precluded in the 14-point assessment. You do not want leads in your schedule. But your long duration task may make strict adherence to the connecting predecessor/successor relationship more cumbersome. Where the joining relationship is FS, in particular, you may be tempted to commence the successor activity before the full completion of the predecessor activity. This is the negative lag modified FS relationship scenario where strict adherence to the FS relationship is not maintained.

One problem with negative lag is its predictive nature; you are deciding to commence a successor activity early based on the predicted future completion of the predecessor task. This predictive nature of negative lags makes them problematic. Keeping tasks within the acceptable 44-day duration makes the prospect of negative lag insertions less likely. So negative lags are forbidden by the 14-point assessment and long duration tasks that make them more tempting are discouraged.

Summary

Long duration tasks are allowed, however, according to DCMA no more than 5% of tasks may be high duration. Long duration tasks require careful inspection. Discrete measurement of long duration tasks is more difficult and less accurate. Long duration tasks may require in-process scope or activity definition changes. Critical path distortion is also a real possibility. And the temptation to insert negative lags is a real problem as they are forbidden by the 14-point assessment.

The DCMA will take the view that the schedule in not being managed well, but rather being manipulated. Implement rolling wave planning to discretely define near term tasks and broadly estimate planning packages of farther out more malleable work efforts. Yes, the high duration assessment supports avoiding schedule pitfalls made more likely by the prevalent use of long duration tasks. Much like high float values, long duration activities/tasks must be routinely identified and assessed as to how better to break them up. This results in a more discrete and robust schedule.

Chapter 10

INVALID DATES

Assessment 9 – Invalid Dates Overview

The invalid dates assessment says that incomplete tasks should not have forecasted (future work) dates in the past or actual (completed work) dates in the future. The threshold for invalid dates is that none are allowed. There are no exceptions.

Assessment 9 – Invalid Dates In-Depth

The status date or data date, depending on your scheduling software terminology, is similar to the closing statement date of your checking account. The closing date on your checking account statement is the latest date through which all checking account activities have been recorded. Likewise, a schedule's status date is the date through which all activity schedule progress has been recorded. An activity's status is denoted in relation to the status date. All completed tasks should be left or prior to the status date. All incomplete or remaining tasks should be right or come after the status date. If your schedule has been properly updated then everything on the left side of your status date should be complete and everything on the right side should be incomplete. This chapter examines the DCMA invalid dates assessment as a measure of the quality of a progressed schedule.

The DCMA threshold for activities with invalid dates is zero. There should be no invalid date tasks. Again, a task has invalid dates if it has forecast (future work) start/finish dates in the past or actual (completed work) start/finish dates in the future, with respect to the status date. This means that the remaining duration of tasks that have not yet begun or completed should be pushed beyond the status date.

Schedules in software like Microsoft Project are more susceptible to invalid dates. This is largely because Microsoft Project does not have a rigid task update procedure. In our Microsoft Project demonstration, Figure 1, task A has both the activity forecasted (future work) start and finish dates come before the status date (or in the past).

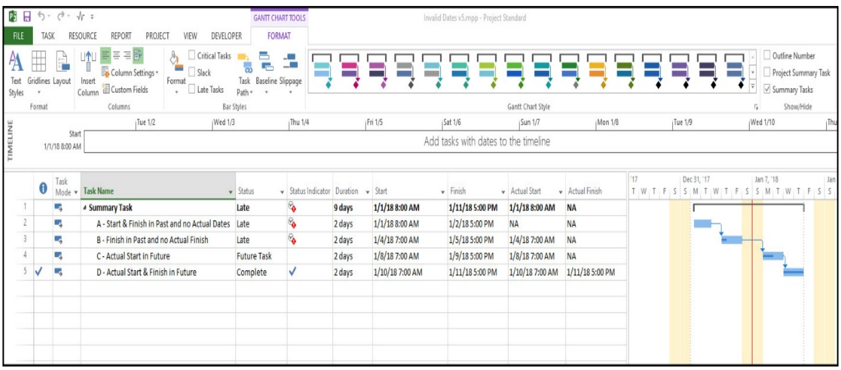


Figure 1

Activity B has the forecasted (future work) finish date come before the status date (or in the past). On the other side the actual (completed work) start of activity C comes after the status date (or in the future). And the actual (completed work) start and finish of activity D is downstream of the status date (or in the future). So in Microsoft Project we see it is possible to have schedules indicating future work in the past and completed work in the future. Schedulers must be more aggressive to make certain that no invalid dates exist relative to their “As of

Status Date”. These violations can be avoided by routinely filtering for all tasks with Actual Start/Completion Dates after the As of Status Date and any Current Start/Completion Dates prior to the As of Status Date.

All of these invalid dates scenarios are possible in Microsoft Project because its update procedures are more fluid. Note, in particular Figure 2, where Microsoft Project schedulers must manually toggle when they want to update the project to reschedule incomplete work to start after a specified date, which may differ from the status date.

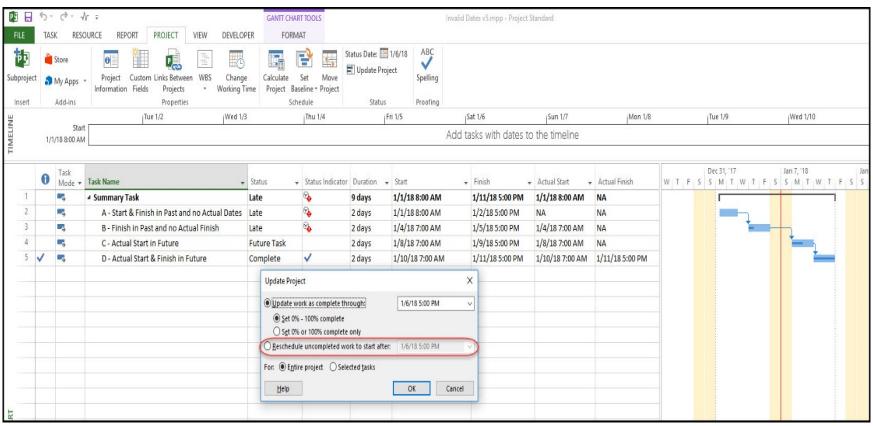


Figure 2

So the shift of incomplete work from the left to right side of the status date is not automatic, which leads to illogical scheduling situations.

Unlike Microsoft Project, however, Primavera P6 does have a well-defined procedure for updating tasks. In fact when the schedule in Primavera P6 is recalculated, the remaining duration for all incomplete activities is automatically moved to the right side of the data date. This helps keep the schedule progress in line with the data date. Primavera P6 procedures are not perfect. It is possible in Primavera P6 to have completed work in the future, i.e. beyond the data date. Figure 2 displays a Primavera P6 schedule where activity B is completed "out of sequence" and in the future, which is not good scheduling practice.

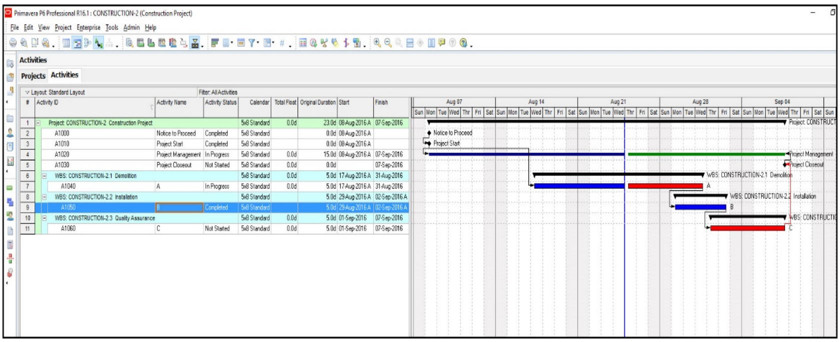


Figure 3

Nevertheless, Primavera P6 is less susceptible to invalid dates.

Summary

Forecasted (future work) start and finish dates that are in the past relative to the status date are suspect. Additionally, actual (completed work) start and finish dates that are in the future relative to the data date are also cause for concern. Both of these situations describe invalid dates. And the DCMA threshold for invalid dates in the schedule is zero. Schedulers should review their procedure for progressing the schedule.

The steps in Primavera P6, in particular, are straightforward:

1. Enter the status of tasks
2. Move the data date forward
3. Recalculate the schedule, which moves all remaining work from the left side to the right side of the data date

Again, for a properly progressed schedule, forecasted (future work) start and finish dates *should not* be left of the data date (or in the past), and actual (completed work) start and finish dates *should not* be right of the data date (or in the future). The status date is therefore the dividing line between task forecasted and actual dates. The DCMA invalid dates assessment rightly seeks to avoid illogical situations were future work is in the past and completed work happened in the future.

Chapter 11

RESOURCES

Assessment 10 – Resources Overview

Resource loading is not a requirement, but if resources are included, according to the DCMA, they must be complete. So, it's all or nothing. When a schedule has resources, all tasks with duration greater than zero in the respective schedule must have either associated dollars or resource hours. Monitor the completeness of resource loaded schedules by calculating the following missing resource metric:

$$\%\ \textit{Missing Resources} = \frac{\textit{Total \# of Incomplete Tasks with Missing Resource}}{\textit{Total \# of Incomplete Tasks}} x100$$

The percentage of missing resources in loaded schedules should be zero.

Assessment 10 – Resources In-Depth

The DCMA resource assessment is somewhat variable. Resource loading a schedule is not always specifically required except in integrated earned value management systems and sometimes as a contractual stipulation. Schedules, however, that are resource loaded should be reviewed to confirm that no resource assignments are missing. All tasks in a resource loaded schedule should either have a resource effort (hours) or fixed price ($) assignment. This chapter considers the DCMA resource assessment as a measure of the quality of a schedule.

Again, the goal is to determine whether a resource-loaded schedule is loaded in entirety or not. A resource-loaded schedule therefore implies that all required labor and measureable material, equipment, and other costs are assigned to appropriate activities. Improperly resource loaded schedules, according to the DCMA resource assessment guideline have budgets that are likely incomplete. Indeed in earned value management systems, all scope, including labor, material and equipment costs must be accounted for through the use of resource assignments to activities.

This resource loading assessment applies to all normal activities and level of effort (LOE) activities (except milestones) that are planned, in-progress, or complete. The DCMA resource assessment specifically states that "all tasks with durations greater than zero have dollars or hours assigned." Some have interpreted this to mean all activities of 1-day duration or greater should be resource loaded.

Others understand that all discrete (measured) effort in the schedule should have work or cost assignments. As mentioned above, milestones which have zero duration are not included. This distinction becomes an issue when one considers the cost loading of, for example, a project initiation meeting. An initiation meeting typically lasts 2-hours. Is that 2-hour meeting a discrete (measured) effort? If so, then the scheduler has to assign all meeting participants as resources, which may become laborious for large meetings that nevertheless have minimal associated costs.
An alternate strategy, particularly for regular meetings is to create level of effort (LOE) management activities that capture all the likely hours that will be consumed in such

meetings. This is preferable and more easily maintained than dozens of weekly meeting activities.

Summary

Resource loaded schedules are not mandatory, but recommended. And, if a schedule is resource loaded, use the DCMA resource assessment to validate that it is resource loaded in its entirety. Do this by verifying the labor hours, material and equipment costs in the proposed budget with those that are being calculated in the schedule. This inspection applies to all normal and LOE activities. It excludes milestones.

It should be noted that there could be a conflict with DCMA guidance regarding their objective to replace lags with what can be called dummy activities. Dummy activities provide insight into what may be happening during a lag between activities, but may have no resources. While this may cause some conflict to DCMA, it is desirable to have the dummy activities rather than lags.

The DCMA resource assessment helps validate the schedule budget, an important measure of project success. And schedules that meet the DCMA resource assessment have a significantly greater opportunity for success.

Chapter 12

MISSED TASKS

Assessment 11 – Missed Tasks Overview

This metric monitors excessive slippage. Calculate the missed tasks ratio as follows:

$$\%\ \text{Missed Tasks} = \frac{\#\ \text{of Tasks with Actual or Forecast Finish Date Past Baseline Date}}{\#\ \text{of Tasks with Baseline Finish Date on or Before Status Date}} \text{x}100$$

Tasks are included in the denominator if their baseline finish dates are on or before the status date. This is the baseline count. Also included in the numerator, are tasks that have an actual finish date or forecast finish date after their respective baseline finish dates. The number of missed tasks should not exceed the 5% threshold.

Assessment 11 – Missed Tasks In-Depth

The DCMA Missed Tasks assessment in particular, measures how well the actual (in-progress) schedule is tracking to the baseline schedule. A schedule that is exceeding its baseline finish dates may be a schedule in trouble. And it may require a corrective action plan to realign the actual schedule and baseline. This chapter reviews the DCMA Missed Tasks assessment as a means for measuring whether the schedule is tracking to the baseline.

It is expected that not everything goes according to plan. Nevertheless your plan is your desired outcome, so you want actual progress to tract the schedule as much as possible. You also want an early warning indicator to alert you when your actual schedule is slipping away from the baseline. The DCMA Missed Tasks assessment is an alert indicator that your schedule requires corrective action to realign it with the baseline.

The DCMA Missed Tasks assessment inspects how well schedule task finish dates are tracking baseline task finish dates. The assessment focuses on task finish dates and not task start dates. Task finish dates are scrutinized because it is more difficult to achieve and report the completion of a task than the commencement of a task. Therefore tracking schedule task finish dates provides a more conservative and/or accurate measurement of schedule progress.

The Missed Task assessment says that no more than 5% of schedule incomplete tasks should have missed baseline finish dates. The missed baseline finish % ratio again is as follows:

$$\%\ \textbf{Missed Tasks} = \frac{\textbf{\# of Tasks with Actual or Forecast Finish Date Past Baseline Date}}{\textbf{\# of Tasks with Baseline Finish Date on or Before Status Date}}\textbf{x100}$$

In simpler terms the numerator is the number of missed tasks and the denominator the baseline count. Note that the denominator baseline count does not include tasks with no baseline dates. Therefore all tasks added after baselining the project are not included in this metric.

So, the assessment inspects task finish dates up through the status date, and allows only 5% of schedule tasks to slip beyond their task baseline finish dates. Otherwise, the schedule is flagged for review. This metric does not measure on-going activities that have a baseline finish beyond the update period. The Missed Tasks assessment is therefore retrospective.

Note that all missed tasks are weighed equally, regardless of their duration and/or cost. The total float of tasks is not considered. It is possible to fail this metric by using available total float, if the early finish dates are missed. Again, this metric is a conservative schedule delay early warning indicator. Also, not at issue is whether or not the activity in question is along the critical or longest path.

Let's walk through a Primavera P6 Professional schedule demonstration. In Figure 1 we have a P6 schedule that has been progressed two months.

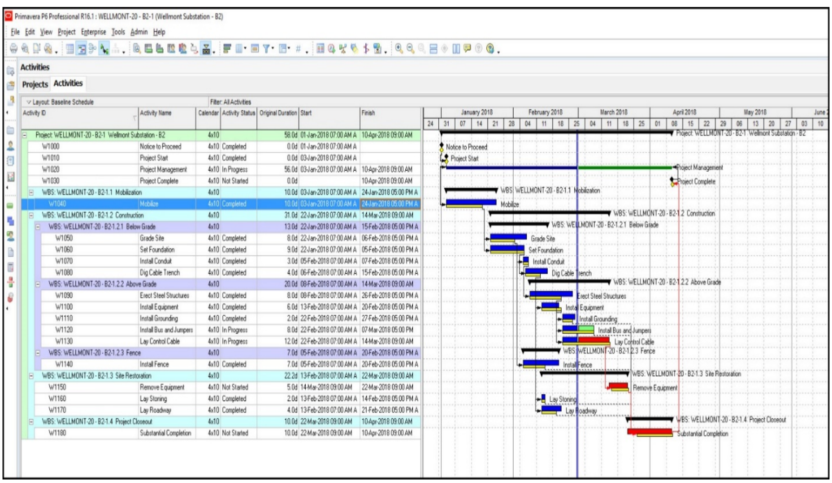


Figure 1

We want to determine the percent of activities missing their baseline finish date, as per the data date.

First we want to find the baseline count. This is the number of activities that are scheduled to complete before the data date. To find the baseline count we implement the filter in Figure 2.

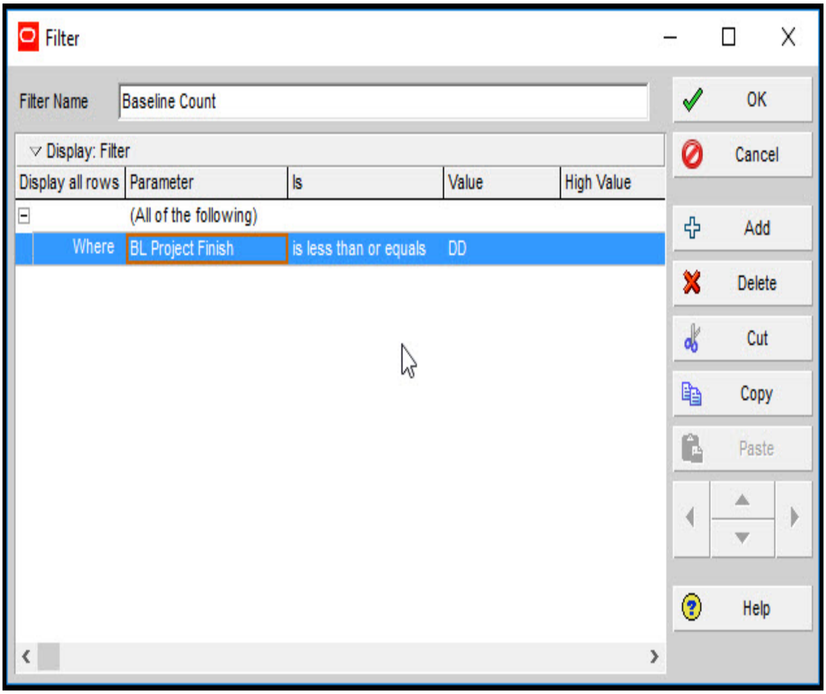


Figure 2

Note we are using the Data Date (DD) modifier to keep this filter focused on retroactive missed dates and not counting the impacted activities out in the future. This filter captures all activities that have a project baseline finish date less than or equal to the DD. The filter results are all the activities displayed in Figure 3.

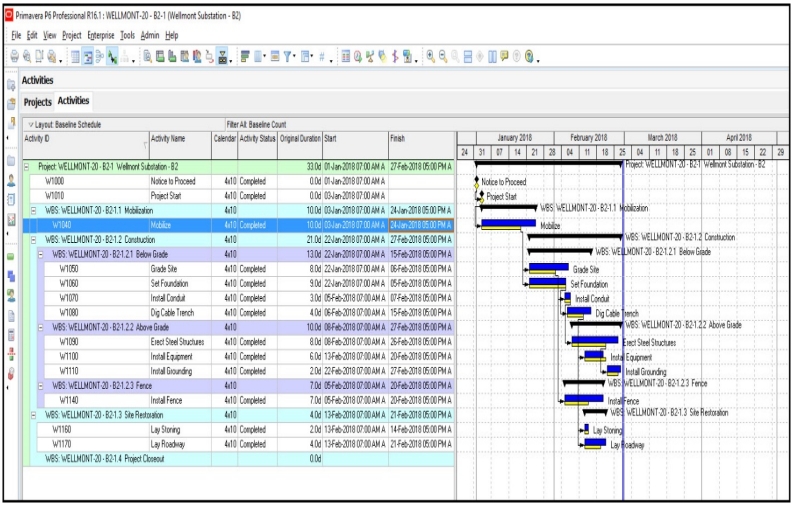


Figure 3

Now we simply count the number of activities displayed. I count thirteen activities that were scheduled to complete before the data date. This includes milestones.

Now we create a missed task filter as displayed in Figure 4.

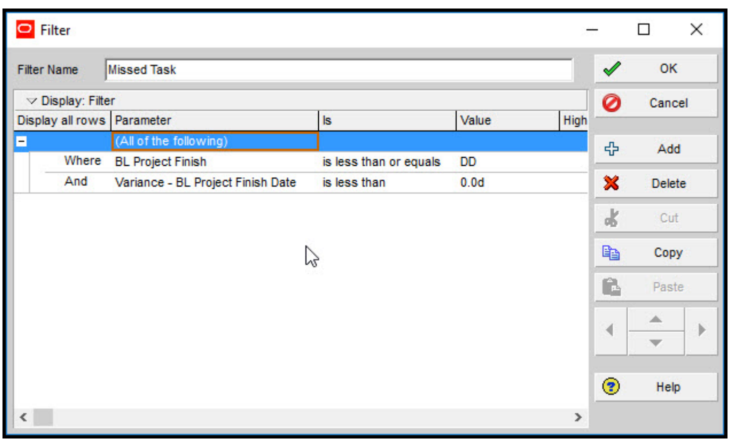


Figure 4

This filter captures all activities that have a project baseline finish date before the data date and have a project baseline finish date to finish date variance (BL Project Finish date minus Finish date) less than zero. All activities that miss their project baseline finish date are captured in this filter. The filter results are all the activities displayed in Figure 5.

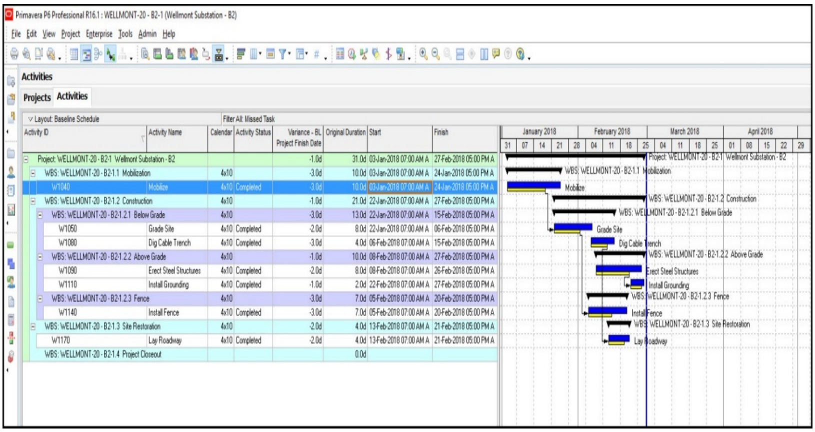


Figure 5

Note the yellow baseline bars in relation to the blue activity bar actuals. Again, we simply count the number of activities displayed. I count seven activities.

So out of thirteen activities that were scheduled to complete before the data date, seven missed their project baseline finish date. We calculate the missed task % as follows:

$$\% \boldsymbol{Missed\ Tasks} = \frac{\mathbf{7}}{\mathbf{13}} \times \mathbf{100} = \mathbf{54}\%$$

So 54% of activities are missing their project baseline finish date, as per the data date. This is not good. Our schedule has failed the DCMA Missed Tasks assessment by a wide margin.

Our schedule requires rigorous changes to realign it with the project baseline. It's Interesting to note how smaller schedules with less activities can fail this test more quickly and severely than much larger schedules. Similar filters can be developed for Microsoft Project schedules.

Summary

The Missed Task assessment provides a conservative and straightforward way to measure schedule progress, and serves as an early warning alert when the schedule is slipping. The metric allows 5% or less of activities to slip their finish baseline dates. The metric does not consider the duration, cost, and total float of missed tasks. This is a retrospective metric. Task project baseline finish dates beyond the data date are not included in the analysis.

So whether or not the DCMA is auditing your schedules, this metric is a useful management tool for tracking the status of any company's project portfolio. We recommend you make this check part of your routine status process to see and report the overall status health of your projects. You may, however, find that this missed tasks assessment is too rigorous or sensitive as even completed activities that barely missed their schedule baseline finish date are captured and/or flagged.

Chapter 13

THE CRITICAL PATH TEST

Assessment 12 – Critical Path Test Overview

Quality schedules have a continuous critical path from the project start milestone to the project finish milestone. It may, however, be difficult to perceive that your critical path is truly unbroken when you have many activities and, possibly, activity constraints.

The critical path test inspects the integrity of network logic and, in particular, the critical path. The critical path test says the project completion date should delay in direct proportion to the amount of intentional slip introduced to the schedule as part of this test. If not, the schedule has broken logic. Broken logic is the result of missing predecessors and/or successors.

Schedules pass the critical path test if the project completion milestone shows a negative total float value or early finish date in direct proportion to applied intentional slip.

Assessment 12 – Critical Path Test In-Depth

How can one be certain that their schedule's critical path forms an unbroken link or chain of tasks from start to finish? Answer: the critical path test inspects the soundness of the schedule's critical path. The ideal, mentioned above, is for one complete critical path from project start to finish. This chapter discusses the DCMA Critical Path Test as a way to guage the soundness of a schedule's critical path.

The DCMA Critical Path Test assesses the integrity of the overall network logic. Schedule's should commence at a single milestone and complete at a single milestone. But in between you may have multiple paths. One of those paths, however, should begin at the schedule start milestone and conclude at the schedule finish milestone. This is your schedule's critical path. It forms an unbroken link through the entire schedule, and it is the longest path through the schedule logic.

The Critical Path Test is implemented by increasing the remaining duration of an open task on the critical path by a specified amount. This should be an open task nearest the schedule start date. Some guidelines recommend a 600-day increase. The amount is not important, but note the slippage value. If there is one unbroken chain of activities from the start point to the end of the schedule, you will observe that the project early finish date is affected in direct proportion to the inserted test slippage. Schedules that have project constraint deadlines will show a negative total float increase in direct proportion to the slippage.

If you do not observe this change to the total float or finish date it means you have a critical path discontinuity (a gap) somewhere in the schedule, and the schedule is flagged for further review. The critical path test is a Boolean pass/fail metric. If a commensurate amount of schedule slippage occurs on the project finish milestone, the schedule passes the Critical Path Test.

A failed test indicates broken logic, which is usually the result of either missing predecessors or successors. The schedule log file in Primavera P6 Professional lists activities without a predecessor or without a successor. Note that only one activity at the beginning of the schedule should be without a predecessor, and only one activity at the end of the schedule should be without a successor.

Summary

Knowing the critical path or longest path through the schedule is important for schedule optimization efforts and to realistically predict the end date of the project. A schedule's critical path should form one contiguous link from the first to last activities. The Critical Path Test provides a simple and effective way to test for critical path continuity. The Critical Path Test is a binary pass/fail test that flags suspect schedules for further activity logic inspection.

Chapter 14

THE CRITICAL PATH LENGTH INDEX

Assessment 13 – Critical Path Length Index Overview

The Defense Contract Management Agency's (DCMA) Critical Path Length Index (CPLI) is a forward looking gauge of the efficiency required to complete a major milestone on-time, most likely the project deadline. A CPLI of 1.00 is acceptable. It says the program must accomplish one day's worth of work for every day that passes. A CPLI greater than 1.00 is good. The project plans to finish early. The CPLI focus is on the achievability of the critical path. Schedules that have a CPLI less than 0.95 are flagged for further review. Let's take a closer look at this interesting metric.

Assessment 13 – Critical Path Length Index In-Depth

The DCMA missed tasks assessment discussed in a previous chapter 12 is an early warning alert that the schedule is falling behind. Whether the missed task assessment warns your schedule is falling behind or not, the CPLI measures the efficiency required on remaining tasks to meet the project deadline. The CPLI indicates the efficiency of the critical path, and is a measure of the relative achievability of the critical path. A CPLI less than one warns that the schedule is progressing inefficiently with regard to meeting the project deadline.

In Figure 1 we have a demonstration schedule.

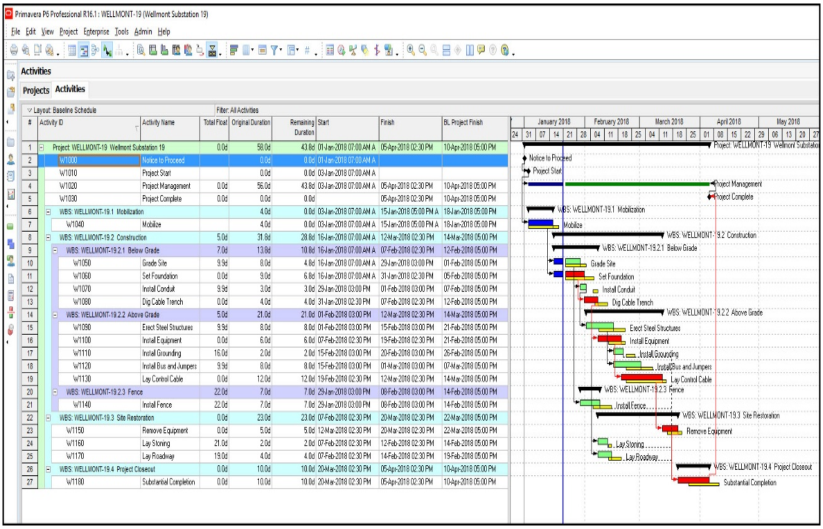


Figure 1

This schedule has progressed two weeks. Note the yellow baseline that indicates we are currently ahead of schedule. Also, the 'BL Project Finish' lists the project baseline finish dates of all tasks. Note, in particular, that the project complete milestone occurs at the close of business on April 5th, 2018. Also, the baseline finish date (BL Project Finish) is April 10th, 2018. Let's create a project finish constraint on this BL Project Finish date, and observe the impact on our total float. Figure 2 displays our schedule with an April 10th project constraint.

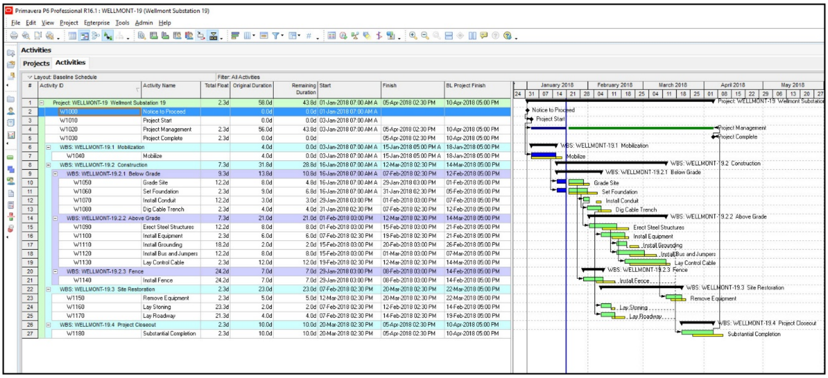


Figure 2

The remaining duration of the project is 43.8 days and the total float of the project complete milestone is 2.3 days. With this data we can calculate the CPLI using the following formula:

$$\textbf{Critical Path Length Index} = \frac{\textbf{CPL} + \textbf{TF}}{\textbf{CPL}}$$

Where CPL is the critical path length and TF is the total float. Our example schedule CPLI computes as follows:

$$\textbf{Critical Path Length Index} = \frac{\textbf{43.8 Days} + \textbf{2.3 Days}}{\textbf{43.8 Days}}$$

The CPLI for our schedule is 1.05, which is good news. The schedule is working along at an efficient pace. If the CPLI was less than one we would know an increased efficiency is required to meet the project deadline. Again, CPLI is a measure of the realistic possibility of meeting the project deadline. And the CPLI assessment flags any schedule that has a CPLI less than 0.95. This indicates a project in trouble.

TENSIX CONSULTING

Summary

The CPLI is a measure of the efficiency required on remaining critical tasks to meet the project deadline. It is a forward-looking measure, unlike the missed task assessment that is retrospective. Data input required for CPLI are simply CPL and TF, which can be found by defining a project constraint. Not only is CPLI a measure of efficiency, but it also warns when a schedule is in trouble of not meeting the project deadline. CPLI values less than 0.95 trigger a flag that the schedule requires further review.

Chapter 15

THE BASELINE EXECUTION INDEX

Assessment 14 – Baseline Execution Index Overview

Many project managers look to Earned Value Management (EVM) values to predict whether or not their project will complete on time. Recently, the Baseline Execution Index (BEI) has been popularized as a schedule completion early warning metric. The BEI metric measures task completion efficiency when compared to the baseline, and relative to the status date. The BEI specifically compares the sum of total tasks completed to the sum of all tasks with a baseline finish date on or before the current reporting period. The BEI formula is stated by DCMA as follows:

$$\textbf{BEIcum} = \frac{\textbf{Sum \# of all Tasks Completed as of the status date}}{\textbf{Sum \# of all Tasks with Baseline Completion dates on or before the status date}}$$

A BEI greater than 1.00 reflects a higher task throughput than planned. Schedules that have BEI less than 0.95 are flagged for further review.

Assessment 14 – Baseline Execution Index In-Depth

The BEI provides both an objective and a nontrivial schedule review that is still sensitive enough to provide project managers an early detection alert, warning that their schedules require attention. Again, a BEI > 1 is favorable and a BEI < 1 is unfavorable. This chapter introduces the BEI as a schedule completion early warning metric.

One of the project manager's major duties is to monitor schedule progress and make appropriate schedule adjustments to keep the project on schedule. The Schedule Performance Index (SPI) is an earned value management metric that helps project managers extrapolate their project's schedule completion situation. The SPI metric divides earned value by planned value (EV/PV) to determine how close the earned value is to what was planned at this particular point in the project lifecycle. The SPI thus provides the project manager keen insight into the progress of the schedule. Recent studies, however, have indicated that the BEI provides an earlier schedule completion warning metric. The BEI is like the *canary in the mine*, providing first warning alert that something is amiss.

Before calculating BEI we want to make sure we understand the formula. DCMA, again, defines the BEI ratio as follows:

$$\mathbf{BEIcum} = \frac{\textbf{Sum \# of all Tasks Completed as of the status date}}{\textbf{Sum of all tasks with Baseline Completion dates equal to or earlier than the status date}}$$

Note: the denominator is the sum of completed tasks and *incomplete* tasks missing a baseline finish date. This is equivalent to total # activities that should have been completed as of the status date. The BEI formula, therefore, simplifies to the following:

$$\mathbf{BEIcum} = \frac{\textbf{Total \# of All Tasks Completed as of the status date}}{\textbf{Total \# of Tasks that Should have been Completed as of the status date}}$$

DCMA wants us to consider the denominator as a *baseline count* equivalent to the number of tasks with a baseline finish date on or before the status date. The BEI becomes:

$$\textbf{BEIcum} = \frac{\textbf{Total \# of Tasks Complete}}{\textbf{Baseline Count}}$$

Where, again

$$\textbf{Baseline Count} = \textbf{Total \# of Tasks with Baseline Finish Date on or Before the Status Date}$$

So the BEI ratio nomenclature is confusing but the formula comes down to two terms: total # of tasks complete and total # of tasks that should be complete (or the baseline count). ***This formula only applies to normal tasks, so we filter out LOE, summary tasks, and zero duration tasks (milestones).***

Again, a BEI > 1 is good and a BEI < 1 is not so good. If tasks are not completing by the status date as scheduled than something is wrong with the schedule. The BEI is a means of measuring task completion issues relative to the status date. If tasks' completions, as of the schedule status data date, is insufficient a BEI < 1 warns the project manager that the schedule is heading off course and needs attention. A key element realized by using the BEI is that all tasks completed as of the status date are included in the numerator. By that I mean that tasks completed that were supposed to have been completed to date plus tasks completed earlier than their planned baseline completion dates are included. This provides the Project Manager with a good appreciation of how well their overall program is progressing relative to their Baseline Plan as of the status date via the utilization of the BEI calculation.

The BEI provides a warning indication that your schedule is veering towards trouble. Note, as an example, a schedule that has 1 task incomplete out of 20 that should be complete by the status date. The BEI computes as follows:

$$\mathbf{BEI} = \frac{\mathbf{19}}{\mathbf{19+1}} = \frac{\mathbf{19}}{\mathbf{20}} = \mathbf{0.95}$$

So this schedule passes the BEI test, which specifies that BEI's 0.95 and above are passing. The same schedule will compute a % missed task from chapter 12 as follows:

$$\% \ \mathbf{Missed\ Tasks} = \frac{\mathbf{\#\ of\ Tasks\ with\ Actual\ or\ Forecast\ Finish\ Date\ Past\ Baseline\ Date}}{\mathbf{\#\ of\ Tasks\ with\ Baseline\ Finish\ Date\ on\ or\ Before\ Status\ Date}} \mathbf{x}100$$

$$\% \ \mathbf{Missed\ Task} = \frac{\mathbf{1}}{\mathbf{20}} \times \mathbf{100} = \mathbf{5}\%$$

It passes the % missed task test that says no more than 5% of schedule incomplete tasks should have missed baseline finish dates. But what if two tasks missed their baseline finish dates, but only one remains incomplete by the status date? In this situation the BEI remains 0.95, but the % missed task ratio computes as follows:

$$\% \ \mathbf{Missed\ Task} = \frac{\mathbf{2}}{\mathbf{20}} \times \mathbf{100} = \mathbf{10}\%$$

So the missed tasks formula numerator includes tasks that missed their baseline finish, but are, nonetheless, complete. The missed task ratio may include minutiae that makes the schedule fail the missed task pass/fail criteria. Is it really a better gauge of progress to include or flag completed tasks simply because they missed their baseline finish date? The BEI simplifies the equation to what's complete by the status date divided by what should be complete by the status date. It's fine for an activity to miss its baseline finish date as long as it is complete by the status date. So the BEI avoids trivial schedule lapses, but provides an objective and sensitive measurement of schedule progress.

It is unfortunate that neither Primavera P6 Professional nor Microsoft Project have BEI variables. Primavera P6 Enterprise Project Portfolio Management (EPPM), however, does include a BEI metric in its schedule check feature. Let's demonstrate a BEI computation in Primavera P6 Professional. We have in Figure 1 our demonstration schedule.

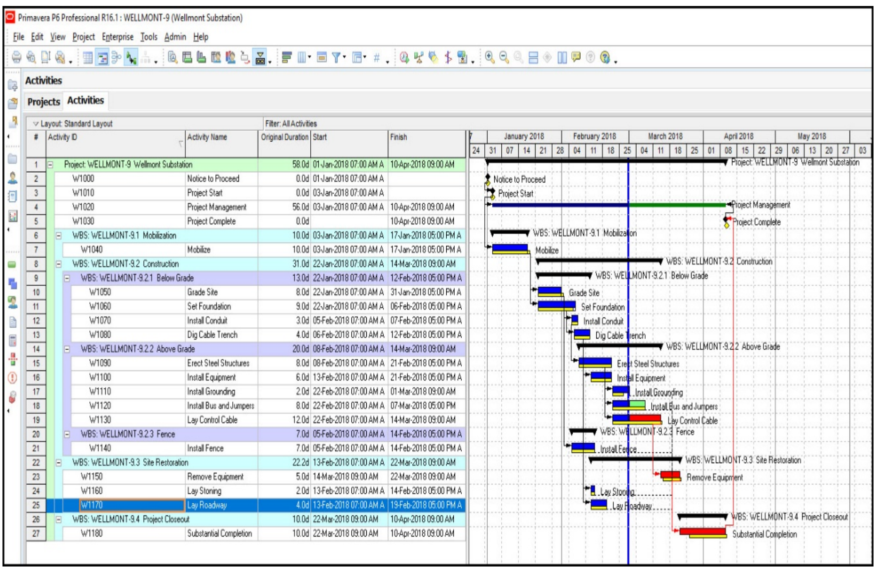


Figure 1

This schedule has progressed 2 months: January and February. The number of tasks with actual finish dates is computed using the following filter, Figure 2:

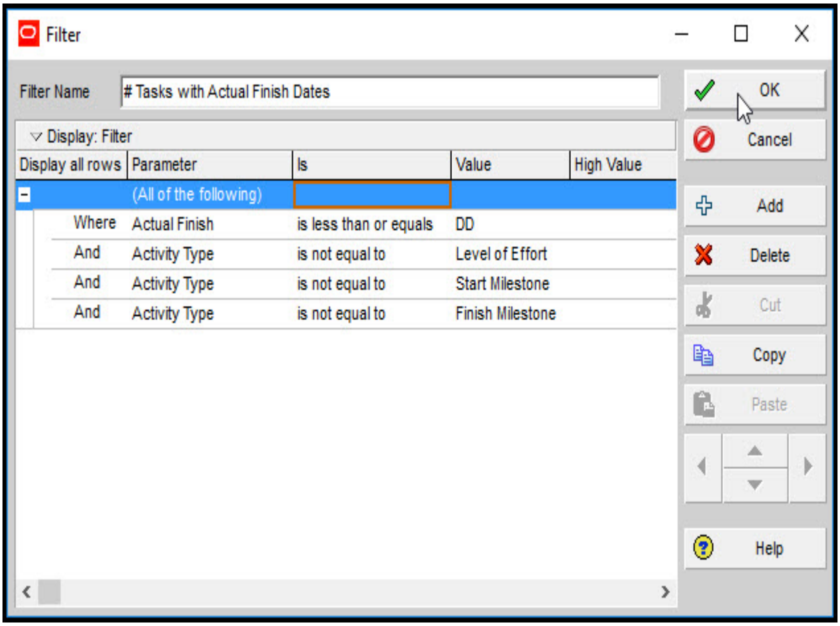


Figure 2

Applying this filter we find that ten tasks have an actual finish date before the status date, Figure 3.

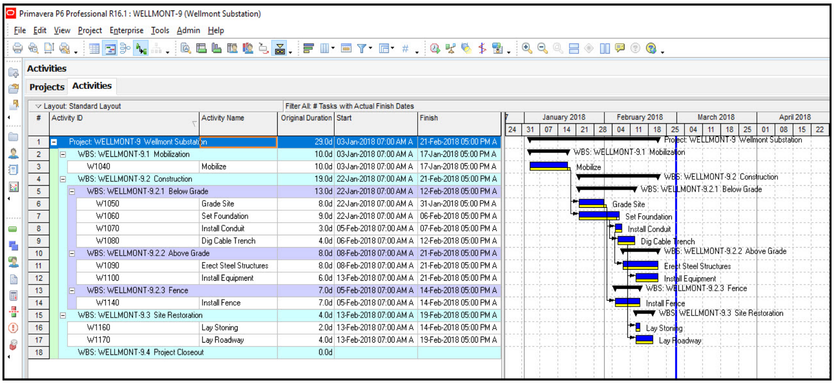


Figure 3

The number of tasks scheduled to finish prior to the status date computes with the following filter, Figure 4.

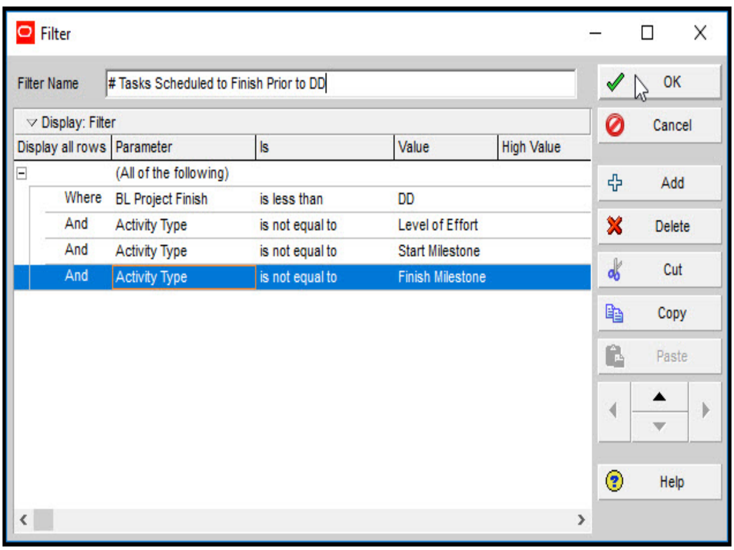
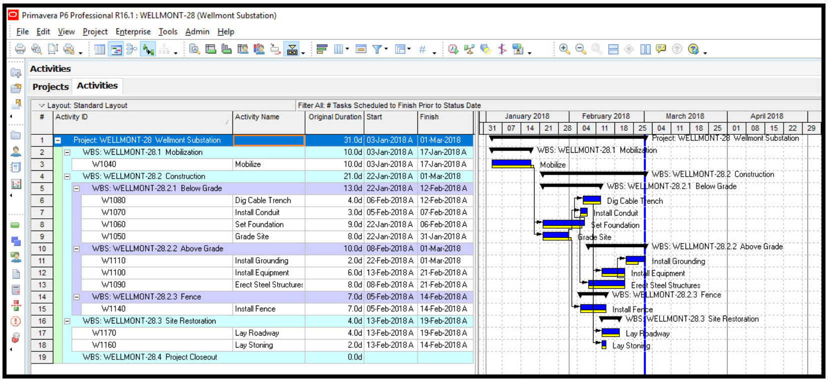


Figure 4

Eleven tasks were scheduled to complete before the status date, Figure 5.



Primavera P6 Professional R16.1 : WELLMONT-28 (Wellmont Substation)

Layout: Standard Layout — Filter All: # Tasks Scheduled to Finish Prior to Status Date

#	Activity ID	Activity Name	Original Duration	Start	Finish
1	Project: WELLMONT-28 Wellmont Substation		31.0d	03-Jan-2018 A	01-Mar-2018
2	WBS: WELLMONT-28.1 Mobilization		10.0d	03-Jan-2018 A	17-Jan-2018 A
3	W1040	Mobilize	10.0d	03-Jan-2018 A	17-Jan-2018 A
4	WBS: WELLMONT-28.2 Construction		21.0d	22-Jan-2018 A	01-Mar-2018
5	WBS: WELLMONT-28.2.1 Below Grade		13.0d	22-Jan-2018 A	12-Feb-2018 A
6	W1080	Dig Cable Trench	4.0d	06-Feb-2018 A	12-Feb-2018 A
7	W1070	Install Conduit	3.0d	05-Feb-2018 A	07-Feb-2018 A
8	W1060	Set Foundation	9.0d	22-Jan-2018 A	06-Feb-2018 A
9	W1050	Grade Site	8.0d	22-Jan-2018 A	31-Jan-2018 A
10	WBS: WELLMONT-28.2.2 Above Grade		10.0d	08-Feb-2018 A	01-Mar-2018
11	W1110	Install Grounding	2.0d	22-Feb-2018 A	01-Mar-2018
12	W1100	Install Equipment	6.0d	13-Feb-2018 A	21-Feb-2018 A
13	W1090	Erect Steel Structures	8.0d	08-Feb-2018 A	21-Feb-2018 A
14	WBS: WELLMONT-28.2.3 Fence		7.0d	05-Feb-2018 A	14-Feb-2018 A
15	W1140	Install Fence	7.0d	05-Feb-2018 A	14-Feb-2018 A
16	WBS: WELLMONT-28.3 Site Restoration		4.0d	13-Feb-2018 A	19-Feb-2018 A
17	W1170	Lay Roadway	4.0d	13-Feb-2018 A	19-Feb-2018 A
18	W1160	Lay Stoning	2.0d	13-Feb-2018 A	14-Feb-2018 A
19	WBS: WELLMONT-28.4 Project Closeout		0.0d		

Figure 5

The BEI computes as follows:

$$\mathbf{BEI} = \frac{\mathbf{10}}{\mathbf{10+1}} = \frac{\mathbf{10}}{\mathbf{11}} = \mathbf{0.91}$$

So this schedule fails the BEI criteria. Our schedule requires further attention. This may be a good time to further inspect the schedule, in particular, its SPI (EV/PV). Note that similar filters can be constructed in Microsoft Project.

Summary

A project manager's major duty is to monitor schedule progress, and make adjustments to keep the project on time. The earlier a project manager can spot and judiciously diagnose a schedule diverting from the plan the better. Recent studies indicate that the BEI provides an objective (nontrivial) but sensitive early warning detection for schedules in danger of missing their deadline. The missed task assessment, however, may be too sensitive to minor schedule lapses.

A variable for BEI does not exist in Primavera P6 Professional nor Microsoft Project. Schedulers may have to rely on output variables and filters to determine both the numerator and denominator in the BEI ratio. Still the early warning the BEI ratio provides, when a schedule is becoming *unhinged,* may be well worth the effort required to compute its value. Again, BEI among progress ratios is both judicious and reasonably sensitive. It provides a clear and definite signal that something in the schedule is amiss.

Chapter 16

SUMMARY

The DCMA 14-point assessment measures schedule quality and alerts schedulers when the schedule has issues that may jeopardize the projects opportunity for success. The 14-point assessment provides confidence that the schedule is a well-modeled and therefore practical schedule. The 14-point assessment's rigorous schedule analysis provides the most thorough review of schedule quality. And having each assessment in mind throughout the schedule creation process supports formulation of a well-built schedule. The following are brief but comprehensive reviews of each assessment to help schedulers keep all the assessments in view.

1. Missing Logic

A scheduler's major effort is the creation of a logically driven dynamic schedule. Schedule logic should hold task dates in place. Schedule tasks missing a predecessor, successor, or both require review. At the least, tie dangling activities into the completion of the project.

2. Leads

Leads are not allowed in scheduling. Leads disrupt the forward flow of the schedule and can be confusing. They also may violate network logic. Leads may be indicative of a schedule lacking sufficient detail. Leads are often replaced by positive lags, but this is not always the best alternative. It's better to model shorter known scopes of work tasks connected by FS relationships, and no lags.

3. Lags

Limit lags to no more than 5% of activity relationships to support schedule clarity. More often, the better solution to positive lag is to define a known scope of work for that particular effort. Consider replacing lags with tasks describing the effort or process, such as cure time. Do not use lags to target successor start dates.

4. FS Relationships

SS and FF relationships are acceptable but consider using FS relationships to simplify the schedule. The relationships assessment says 90% of schedule dependencies should be FS. Other relationship types are more difficult to trace and understand. Therefore, instead of SS and FF positive lag customized relationships, connect known scopes of work tasks with FS relationships.

5. Hard Constraints

The hard constraints assessment says to limit their usage to 5% of uncompleted tasks. But hard constraints should be avoided as they may violate network logic. Constraints of any type are discouraged as they render the schedule static and inflexible to updates.

6. High Float

Every project manager wants positive total float on their tasks. But too much high float may indicate a missing successor or incorrect logical linkages. Review tasks that have greater than 2 months total float and limit their usage to 5% of incomplete tasks.

7. Negative Float

Schedules that have negative float tasks are already behind. Any schedule that has negative total float should have a corrective action plan. Also, document the negative total float with a note of explanation.

8. High Duration Tasks

Limit long duration tasks to 5% of incomplete tasks. Keep task durations within 2-months to support schedule updating and reporting efforts. Exceptions include long duration procurement activities and level of effort tasks.

9. Invalid Dates

Forecasted (future) work should not be in the past and actual (completed) work should not be in the future. The threshold for invalid dates is zero. This is to avoid illogical situations where future work is planned for the past and completed work happened in the future. The status date is the dividing line between actual dates and forecasted dates.

10. Resources

Resource loading is not a requirement. However, resource loaded schedules must be completely loaded. This means all activities except milestones must have a cost or associated resource.

11. Missed Tasks

The missed task assessment monitors excessive slippage. Only 5% of activities can slip from their finish baseline dates. Note that this metric is a conservative and retrospective measure of schedule progress. It may, however, be too sensitive to minor schedule lapses.

TENSIX CONSULTING

12. Critical Path Test

This assessment ensures the schedule has one continuous linkage from project start to finish. Schedule discontinuities are highlighted by the critical path test. Schedules failing this assessment require further review to inspect activity logic.

13. Critical Path Length Index

The CPLI is a forward looking gauge that assesses required efficiency to complete the project on schedule. It also warns when the schedule may miss the deadline. Schedule's that have a CLPI of less than 0.95 require further review.

14. Baseline Execution Index

The BEI is a judicious and important early warning indicator that a schedule is in trouble of not meeting the deadline. Most likely your scheduling software does not have a BEI variable, so you may have to compute the ratio yourself or purchase an additional scheduling software supplement. But the BEI ratios advanced, nontrivial, and purposeful warning makes the computation worth the effort.

Concluding Remarks

The DCMA 14-Point Assessment is a rigorous schedule quality inspection guideline. But some of the assessments are not quite rigid enough. An example of this is the hard constraints assessment. Really, sound schedules would not have any hard constraints because they can violate logic. However critical hard contractual dates cannot be avoided. Hard constraints should be limited to minimize the impact on schedule logical analysis. This would jeopardize the integrity of the entire schedule. As noted, the 14-point hard constraints assessment allows up to 5% of activities to have hard constraints. On the other hand, the missed task assessment, saying only 5% of activities can slip from their finish baseline dates, may be difficult to achieve. Only one in twenty activities can slip.

What the assessment does well, is to ensure that you have a dynamic schedule that is well documented. This is a schedule that can be understood not only by those who created it, but also by other, possibly, less informed stakeholders. One also wants to make sure the schedule is logically complete; that it has no missing predecessors or successors. And it’s important for cost loaded schedules to be fully resource loaded, in that all tasks have either a resource effort or a fixed price. The resource assessment helps confirm that cost and resource loaded schedules are completely loaded.

Further, the assessments alert project managers to schedules in trouble and, ratios like the CPLI, provide guidance towards reaching a successful project completion. Several retrospective and forward looking assessment metrics provide early warning detection when the schedule is in danger of not meeting the deadline. The BEI, in particular, provides probably the earliest consequential warning detection of schedule completion issues.

The 14-point assessment centered scheduler will additionally avoid common pitfalls that could inhibit quality schedules. This includes illogical schedule progression. Yes, creating the project plan is only part of the scheduling effort. You must also progress the schedule in a way that makes sense.

You can audit your own schedule using tools such as Deltek Acumen, Steelray or the aforementioned Primavera P6 check schedule feature to see if your schedule will pass a DCMA 14-point assessment. These software tools provide a comprehensive analysis of schedule quality. They are also helpful for quick and comprehensive schedule inspection. But, again, studying to understand the principles behind the assessments is recommended as this will provide helpful guiding knowledge to the scheduler from project inception.

TENSIX CONSULTING

tensix.com