SOCIETY OF CONSTRUCTION LAW

Endorsed for

TÜRKİYE

10th

INTERNATIONAL SCL CONFERENCE

2023

www.sclturkey.org

DAILY WINDOWS ANALYSIS REVISITED

Thomas C. Long
Director, Time-Logic Inc., Delray Beach, Florida, USA

ABSTRACT

When using a windows based forensic schedule analysis, the durations of the time windows are often thought to be limited in their accuracy, to the interval of the schedule updates. This is due to the fact that between updates, the schedule has not been progressed. So, it is intuitive to conclude that any schedule calculation made with smaller intervals than the progress updates, must be some form of an estimate or proration that exceeds the ability of the update information, to accurately provide.

This, however, is a fundamental misconception, that is rooted in the manner in which the industry updates its schedules for planning and cost purposes. A closer inspection of this, reveals that the optimal method of updating schedules for planning, can be different from the optimal method for quantifying impacts using a windows based forensic delay analysis. However, this disparity only arises in updating the activities which are in-progress at the time of the update.

Once this inconsistency is removed, the equation used to progress the schedules becomes both objective and linear, allowing the schedules to be accurately calculated on a daily basis. The results of these daily schedule calculations, can then be tracked and aggregated through time, in order to capture trends of delay and acceleration, as well as shifts in the critical path, which occur between updates and could otherwise be unobservable.

This paper will detail how updating schedules optimally for planning and cost purposes, can give misleading results with regard to quantifying impacts. Next, it will detail how schedule calculations can be modified to address this discrepancy. Finally, it will show how the recalculated schedules can be used to implement a daily window analysis, which can be analyzed using artificial intelligence and animated graphic techniques to pick up trends, document impacts and even forecast delays to a project, often weeks before their impact.

1. BACKGROUND

At the time of writing this article, updating a schedule on large projects, is typically done once a month (Society of Construction Law, 2017). And when this update takes place, activities that are in-progress (started but not completed) at the time of the update, are assigned a remaining duration based on an estimate of how much work remains, and how long it will take to complete this work (AACE International, March 19, 2018 revision). So, in short, the remaining durations of these activities are based on the remaining work and not the remaining time of the original duration. Expressed as an equation this would take the following form:

D(r) = D(o) * (1-PC)

Where:

D(r) = Remaining Duration

D(o) = Original Duration

PC = Estimated Decimal Percent Complete of the Work (Todd Dunn, 2005).

When using this equation, the actual duration has no part in the formula, and the planner is changing the units of measure from the remaining time, to an estimate of remaining work. In doing so, this could be projecting a delay, before the time given to do the activity is exhausted. If one quantifies the progress based on time (specifically the amount of original duration that remains), then no such projection occurs (Long, 2023). At first glance, using linear progress toward completion of an activity based on how much time has expired (compared to the original duration), seems a short cut (Todd Dunn, 2005). It is not intuitively easy to understand how it could yield a more precise quantification of delay compared to using an estimated remaining duration based on the remaining work. But this deserves a closer inspection, to see how that can occur.

Take the following example of the progress of an activity. Suppose an activity in a schedule has an original duration of 10 days. After 5 days, 50% of the original duration has expired, but only 10% of the work has been accomplished. Updating by remaining work would yield a remaining duration of 90% or 9 days. Updating by remaining time (remaining original duration), would yield a remaining duration of 50% of the original duration or 5 days. It seems common sense that if there is 9 days of work that is remaining, that using a remaining duration of 9 days, is a more precise method to use for planning and cost purposes… and it is. But it is generally a less accurate method when quantifying delay. When quantifying delay, using remaining duration based on the remaining original duration, can yield more accurate and objective results. This is because using remaining work can project a delay before one has actually occurred (Long, 2023).

In the above example, by utilizing remaining work to estimate a remaining time of 9 days, we have projected a 4-day delay to this activity. This projection is made 5 days before the time given to complete this work has expired. It is conceptually similar to a class assignment due at the end of the week, and then an instructor marking the assignment late because you are behind schedule, but before the paper is even due. This analogy is relevant to delay analysis, because the task could be accelerated and finished on time. It has, in essence, been prematurely assigned as late. This becomes problematic if another equally critical activity does not manage to finish within the time given (original duration), but critical delay has already been assigned to an activity that ultimately finished on time.

It is not being suggested that monthly updated progress schedules be performed based on remaining original duration, rather than remaining estimated work. It is vitally important to planning that an accurate as possible estimated remaining duration be utilized for cost controls and as a barometer of progress.

But for quantifying delay, it is important to use the remaining original duration for in-progress work. In this way, delays begin to aggregate once the time given to complete a task has expired. To make this point clear, taking the above example. Assuming that as projected, the activity actually does finish 4 days late. Using remaining duration based on time, would start accruing the delay, once the original duration expires. So, instead of projecting the activity as late, once the work was behind schedule, the first day of delay would be on day eleven, which would compile day for day until the activity finished four days late, just as common sense would dictate.

2. MODIFIED SCHEDULE UPDATING FOR DELAY ANALYSIS

There is another major benefit to updating based on remaining time; progress is linear and calculations can be performed to objectively update the schedule to a single unit day. When updating based on remaining time, the subjective calculation of estimated work is removed, and the actual duration is introduced. This makes the equation both objective and linear, because the original duration is fixed, and the actual duration progresses linearly. If expressed as a formula, it would take the following general form:

D(r) = D(o) – D(a)
Where:
D(r) = Remaining Duration
D(o) = Original Duration
D(a) = Actual Duration

This computation only requires objective information from time-oriented variables that progress in a linear manner, allowing calculations to be made daily.

3. DAILY WINDOWS ANALYSIS CONTINUITY

With information on progress broken down to the day, the activities responsible for controlling completion can be identified and their impacts can be quantified, aggregated and tracked day by day.

To show how this method can reveal a continuity between updates, the following real project example is shown as figure 1. The graphic shows the daily windows analysis for a rail project updated daily according to the monthly progress schedules, only using remaining duration based on remaining time. The graphic's x-axis shows the date of the project, and the y-axis shows the completion date on the left, and the days of delay on the right, both having a one-to-one correspondence. So, accordingly, the positive slopes show delay and the negative slopes in the line graph denote recoveries. Notice that in the month of December, the controlling activities responsible for delay changed 8 times in the first 10 days, and the controlling activity in beige which ended the month, and controlled completion for the longest duration, was not responsible for any delay on the project. This is a fact that could have gone entirely unnoticed using the typical monthly progress windows analysis based on remaining work.

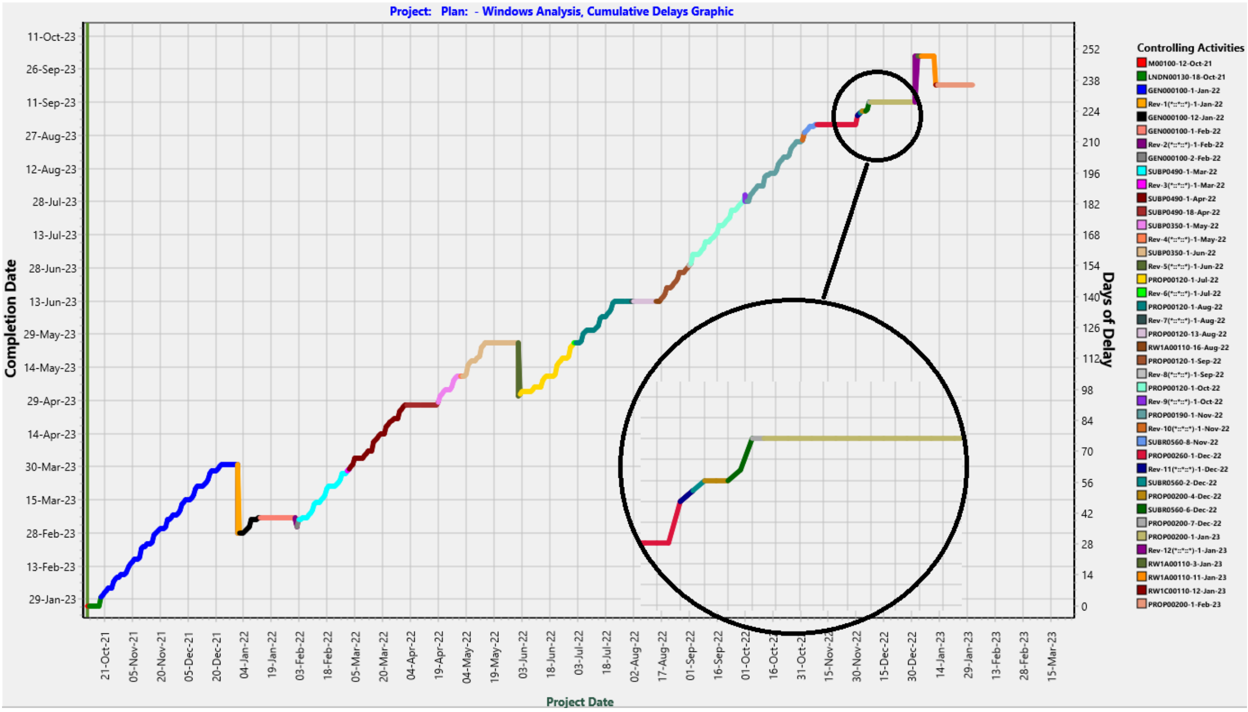

Figure 1. Cumulative delays that detail rapid changes in controlling activities between updates.

Changes in the critical path, do not just occur at the end points of a monthly update. They usually accumulate along the way. It takes a rigorous inspection to flush out delays between these points, if you are using subjective estimates of remaining work of in-progress activities. Using the remaining original duration, however, reduces this to an objective calculation.

It is important to also quantify how changes to the schedule affect the overall monthly impact, in isolation from the impacts to the progress of the work. The impact of these schedule changes can also be flushed out using a daily windows analysis, by utilizing what is known as a "half-step" method, or in terms of AACE taxonomy, an MIP 3.4 schedule analysis (AACE International, April, 2011 revision). In this method, the progress from the latest schedule is put into the previous monthly update. Next, the previous monthly update is updated every day, including on the day of the next monthly update. This then isolates the impacts due to the progress of the work, from the impact of the revisions to the plan. The vertical bars circled in the following figure, denote the impact of plan changes at the end of the month. By looking at the critical path of both schedules, updated to the same data date, one can isolate the changes responsible for the impacts of revisions to the plan. These changes can be isolated computationally, without the need for time consuming inspection.

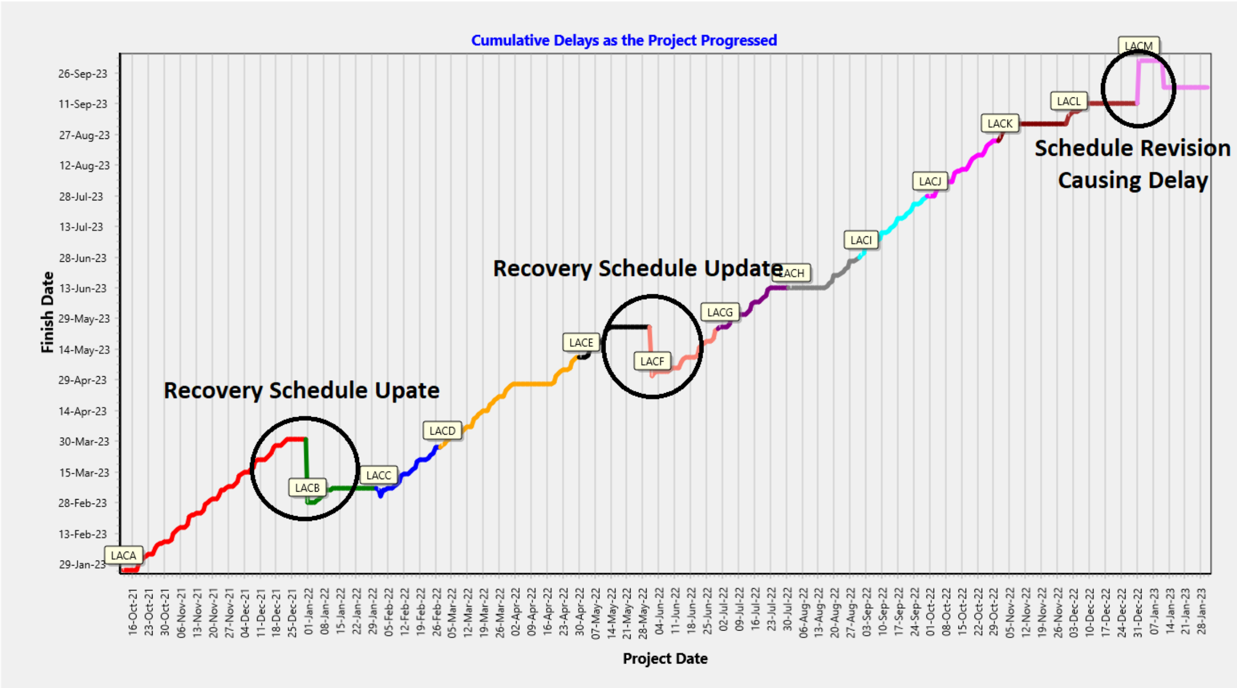

Figure 2. Cumulative delays through time, illustrating schedule revision impacts in isolation.

4. COMBINING DAILY WINDOWS ANALYSIS WITH NEW COMPUTER METHODS

A. Contextual

Although daily windows analysis has been the subject of various articles over the last two decades, and it is specifically referenced in the AACE © International, as a genre of forensic delay analysis (AACE International, April, 2011 revision), it has traditionally taken a lot of resources and computer time to execute properly. But recent advancements in computer technology have dramatically brought down the time and resources necessary to execute this analysis, as well as enhanced its capabilities.

In this last section, a brief overview of these enhanced computer techniques will be reviewed for non-programmers as to its unique compatibility to enhance the speed and insight that each contributes to a daily windows analysis methodology.

B. Artificial Intelligence

The recent release of artificial intelligence open-source libraries such as OpenAI, makes deep learning capabilities, accessible for the general public to use in their software. The techniques are free to incorporate into anyone’s software code currently without charge (except the token purchase necessary for larger consumers of the data).

This brings an especially powerful augmentation to the capabilities of daily windows analysis. There is not much information for insight into delays, if you are only providing the AI with

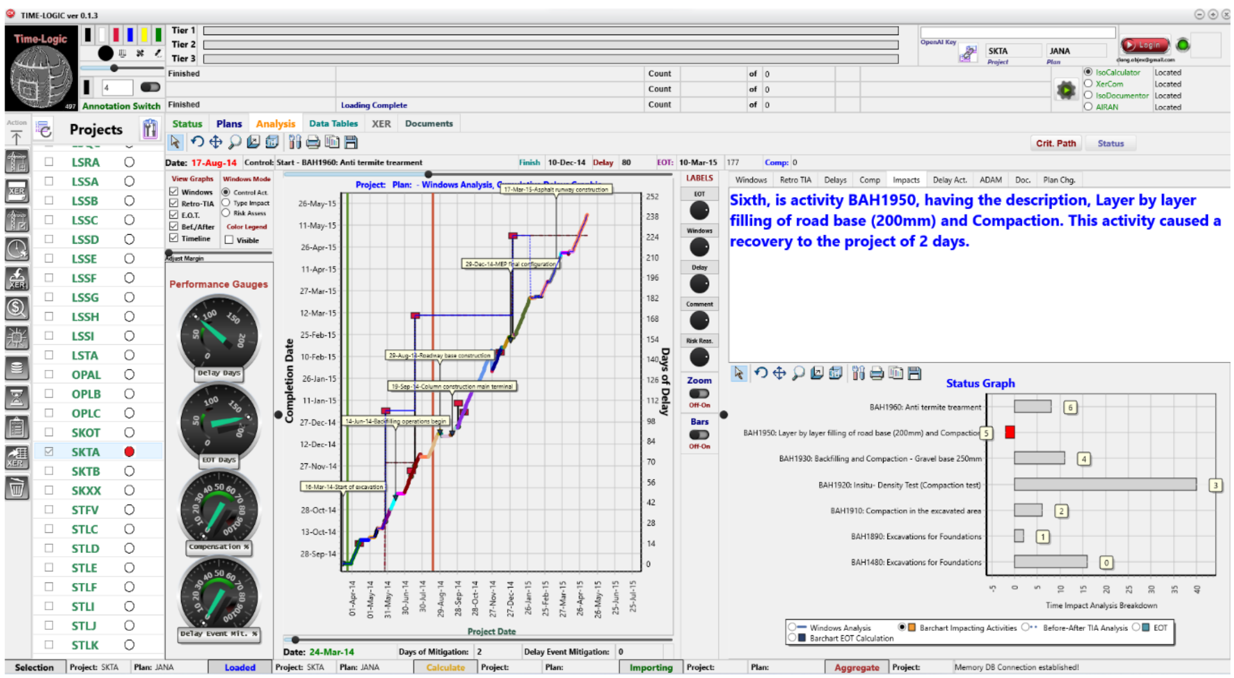

Figure 3. Shows an AI analysis of daily windows tables.

sporadic monthly update information. There is not enough continuity to analyze trends when updates are a month apart. However, ChatGPT and similar chatbots can take information not only from internet data sources, but it can specifically take information in text format that is provided to it, by an application.

Because there is so much information in the database tables of a daily windows analysis calculation, it can provide enough data to recognize trends which users on their own, may not be able to put together, and relate them to the user in plain English.

The way these systems work is on a brute force method with access to massive amounts of information. Most use a neural network, which basically looks at the initial conditions and final outcomes over a vast amount of data. It works to an answer, and depending on the accuracy of that answer, it adjusts the weights of the initial conditions in a feedback loop, over and over again, until it can reasonably predict an outcome without knowing the nature of the process it is observing.

The daily windows analysis contains tables with the daily completion dates, controlling activities, their relationships to delay events, their daily impacts both delay and recovery, as well as timeline information. It can also store documentation in pdf format, such as daily reports, claim and contract information. All of these can be compiled into delimited text that ChatGPT or other similar chatbots, can take as initial conditions, to provide sometimes deep insight into relationships between the data.

As an example, one could give the AI the daily progress information and controlling activity IDs from the daily windows analysis of the P6 progress schedules. This could be supplemented with other pdf files such as a claim documents from a contractor. Once this initial information is fed into the chatbot, one could ask it if there are correlations between delay events in the claim and the controlling activities of the daily windows analysis.

This could be done by writing code to convert the pdf into text, then stream it to the chatbot. Similarly, code could be written that can take the data from the daily windows tables and convert it to delimited text and also stream it to the chatbot to produce an accurate answer saving time and resource delving through large documents and data tables.

One could ask the AI to scan the claim and create tab delimited dates and one sentence synopsis of important events. That information could be parsed and input into a timeline table, which the software could display along side the daily windows analysis graphic.

But these capabilities can even get more intense, as once the AI and its libraries are incorporated into the software, a programmer could actually ask the AI to produce the code that generates text from a pdf, or text from tables, or answers from the chatbot back into the tables of the daily windows analysis. Once incorporated into the software, the AI can generate better more capable versions of the software it is embedded in. This is not a future capability, but one that is available now and is already being put to use in delay analysis software. It is a sea change in the capabilities of software to analyze schedules.

C. Parallel Programming

While AI can enhance the capabilities of a daily windows analysis, parallel programming can greatly increase the speed of these calculations by several orders of magnitude. Parallel programming is a technique which maximizes the utilization of the core processors, or the brains of the computer. Each core of a computer can be seen as analogous to a human brain, as it is the main processor of information. Each core generally speaking has two threads. The two threads of a core can be seen like the right and left hemispheres of the brain. Each of the threads are capable of independent calculations.

Parallel programming allows the programmer to create loops that instead of going sequentially through calculations, it will do as many calculations at once, as there are threads available. This only saves significant time, if the calculations are not dependent upon each other. With daily windows analysis, each calculation is separate and independent for that day. And it is only when all of the calculations are complete, that they are aggregated together to find the trends in the analysis. This makes using parallel programming, ideal for calculating a daily windows analysis. If you have 64 cores available, you can calculate 128 days at one time.

D. Tethering

Tethering is a system where two different applications share data with each other. Unlike a typical server whose architecture has a client-server framework, tethers use a peer-to-peer communication. This means that each application can ask for information, and process requests from and for another application. When one hooks up earphones to a cell phone, it is done with a tether, which in that case is performed through a blue tooth connection. But with delay analysis applications, the tether is mostly between two applications on the same local area network. This is relevant to daily windows analysis because it can free up the end user’s computer from the heavy calculations made using parallel programming. A tether will first search the current computer, but it will also ping the other computers in the network to see if the application it is looking for, is available. Once the companion application is found, then the information to calculate a daily windows analysis can be transferred to the other application, which could be another work station on the network, dedicated to performing the daily windows analysis calculations. In that case, while the work station is bearing the burden of the parallel core usage, the main application's user can be free to handle other tasks. Once the work station is through making the calculation, the tether can be regenerated to transfer the calculated project back to the main user's computer, to analyze the data.

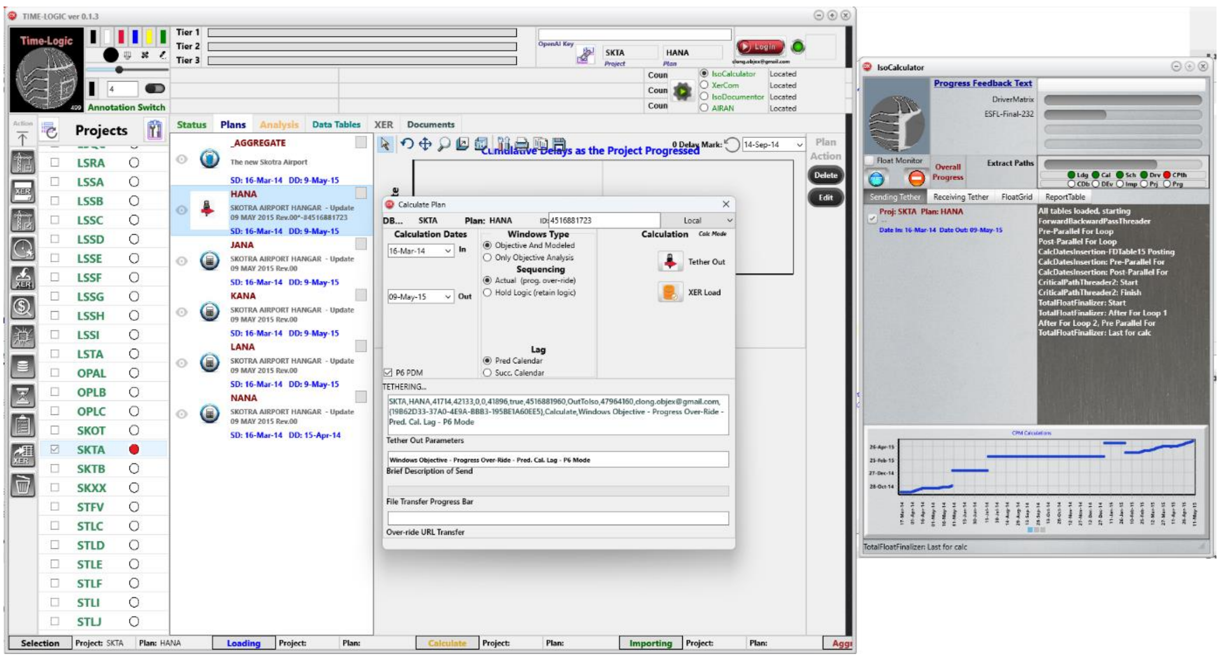

Figure 4. Shows two applications tethered together. On the left, the main delay analysis software, on the right is a separate application which performs the calculations using parallel programming.

5. CONCLUSION

The detail provided by using a Daily Windows Analysis creates a continuity which in conjunction with newer computer methods, such as deep learning AI, allows insight into delay processes and trends which occur between the more typical monthly windows analysis. Furthermore, parallel programming and modern software architecture can dramatically increase the speed of the calculations, making it far less time-consuming and cost-prohibitive than they have traditionally been.

However, typical monthly updates are still performed calculating remaining durations based on estimated remaining work alone, and Daily Windows Analysis relies on a remaining duration based on remaining time. This can lead to differences between the critical paths and the quantification of delay derived using the monthly progress update, compared to those calculated using a Daily Windows Analysis. This, in turn, could undermine a tenet of the contemporaneous windows analysis, namely, that the parties were aware of and working under the assumption of the critical path revealed in the monthly progress update.

If it became a common practice to analyze monthly updates based on remaining time, as well as remaining estimated work, this disparity could be eliminated and new insights on delay developments provided to all parties.

REFERENCES

AACE International. (April, 2011 revision). *Recommended Practice No. 29R-03, Forensic Schedule Analysis.* Morgantown, WV, USA: AACE International.

AACE International. (March 19, 2018 revision). *Recommended Practice No. 90R-17, Statusing the CPM Schedule - As Applied in Construction.* Morgantown, WV, USA: AACE International.

Long, T. C. (2023). CDR-4082: Schedule Updates Using Remaining Original Duration versus Remaining Work Duration. *AACE International 2023 Conference* (p. 13). Chicago: AACE.

Society of Construction Law. (2017). *Society of Construction Law, Delay and Disruption Protocol.* Leicestershire: Society of Construction Law.

Todd Dunn, P. (2005). Strategies for creating accurate updates of CPM construction schedules. *American Society for Engineering Education Annual Conference & Exposition.* Portland.