# *Collapsed As-Built: The Right Way*

*by Kenji P. Hoshino*

The purpose of this presentation is to respond to the misguided and unwarranted criticisms of the Collapsed As-Built method and to provide as set of guidelines to correctly implement the method.

## I. Introduction

The basic theoretical and methodological bases of Collapsed As-Built (CAB) method have been presented in a previous paper published in 1995 for a Project Management Institute Symposium. In that paper, the CAB was described as follows: **(Slide 2)**

> *The Collapsed As-Built CPM Schedule Analysis is a comparative method of quantifying the compensable delay duration by utilizing actual progress data rather than theoretical or planned activity progress. By avoiding the introduction of estimated durations, the issue of the reasonableness or the feasibility of the Contractor's theoretical plan is circumvented. Instead, the basis of the analysis rests on the as-built schedule which is based on actual progress. Critical project delay is measured by comparing the actual, as-built project completion date with the completion date calculated on the Collapsed As-Built CPM Schedule. In this context, the Collapsed As-Built is defined as the residual schedule created by removing all occurrences to be measured from the As-Built Schedule. The analysis is well suited in evaluating the legal proposition: But for the problems at issue, when could the project have completed? One of the advantages of the method is that the Owner is not charged for any Contractor-caused delays, inefficiencies, and weather impacts since they are not extracted from the As-Built and thereby become built into the Collapsed As-Built schedule. Viewed another way, this means that the effect of concurrent delays become factored into the final conclusion by the CPM logic, eliminating the need to perform a separate concurrency analysis. This keeps*

> *the parties focused on the central issue of liability for the problems at issue rather than on esoteric CPM calculation theories.*

For additional information on the basics of CAB, please refer to the attached copy of the paper, Collapsed As-Built CPM Analysis.

# II. Response to Common Objections

Because of the relative infancy of the field of forensic application of the CPM, fallacies and misunderstandings abound. And because of the adversarial context in which forensic analyses are developed, the misunderstandings become the bases for hastily formed objections and fallacious criticisms. In turn, these criticisms become part of the repertoire of opinions that are offered to cast doubt on analysis methods other than the one the critic is utilizing to arrive at his or her conclusions.

When one cuts through the partisan positioning, the truth is that there is no method that is inherently wrong. An analyst may err by incorrect implementation or fail by incorrectly proposing the method to support a concept for which it is not intended, but all methods are good for something. Conversely it is also true that there is no analysis method that is immune from manipulation and abuse.

To date, none of the criticisms and objections to the collapsed as-built method properly addresses any theoretical flaw. None of the objections I have seen to date are based on a properly implemented CAB analysis. The criticisms are usually targeted, unwitting or not, at poorly implemented examples of the method. 🖫 **"Theory vs Implement" (Slide 3)** Often, they reveal the critic's lack of understanding of the method. Listed below are some of the common criticisms borne out of misconceptions about the method. They were collected from various published sources and seminar outlines I have collected over the past five years. They can be categorized into three groups: 1) Attacks against the as-built model, 2) Attacks against the collapsed schedule, and 3) Attacks on the process.

## A. Attacks on the As-Built Model **(Slide 4)**

**1. "Ignores How the Project Was Actually Built" (Slide 5)**

Obviously the only schedule that shows actual construction logic is the as-built. By using the as-built schedule as the starting point, the CAB method is the only comparative analysis method that specifically recognizes actual construction logic. Other methods use the baseline schedule as the starting point and, by definition, ignore actual construction logic.

**2. "Assumes That the Baseline Schedule Logic Was Followed" (Slides 6-12)**

This particular criticism was based on the definition of the methodology as "an analysis in which as-built activity durations are input into the baseline schedule to determine critical path delays." What is described here is not a collapsed as-built. The primary feature of a properly modeled collapsible as-built is that it does not rely on the baseline schedule logic. Therefore this criticism must have been directed at a poorly implemented model or is a result of mistaken identity.

**3. "Assumes That No Delay Mitigation Took Place" (Slides 13-14)**

This criticism is a variant of the previous one and is based on the same mistaken premise that the baseline logic was used to model the as-built. A correctly constructed as-built model takes into account all the mitigation that took place. In fact, the mitigation built into the as-built model often results in a collapsed as-built that has a shorter overall duration than the baseline. This is often used to argue the Contractor's 'right to finish early'.

**4. "CAB Does not Identify the As-Built Critical Path" (Slides 15-20)**

It is true that the CPM model of the as-built, in and of itself, will not give any computed delineation of the as-built critical path. By definition, an as-built schedule carries no float because as-built activities have fixed start and fixed finish dates [1]. This is not to say that a path of controlling items or work cannot be determined through the examination of the documents and the monthly updates. Also, after subtracting the delays at issue, the CPM model does delineate a

[1] Recall that calculated float is the mathematical difference between early and the late dates, which no longer exit in an as-built. Because the technical definition of the critical path is the path carrying the lowest calculated float value, it is only logical that there is no calculated critical path available.

calculated critical path, which can be applied to the as-built as the ***Analogous Critical Path*** [2] to perform the reconciliation calculations for compensable delay.

**5. "CAB Ignores the Fact That the Critical Path is Dynamic" (Slides 21-25)**

The dynamic nature of the critical path cited in this criticism refers to the fact that while the project is being built, the critical path may not follow the critical path delineated in the baseline schedule, but will shift from path to path depending on reported progress and logic changes inherent in any operation. Some critics also call this the 'single unchanging critical path' fallacy. It is important to understand that the critical path is dynamic only during the project when the schedule still carries incomplete activities. Once the project is complete, the as-built schedule carries a path of controlling items of work, which closely follow the series of critical one-month windows created by the monthly updates. But the hindsight as-built path defined by these successive critical paths traced by the updates is static, by definition. If that fact is called 'freezing the critical path', so be it. There is nothing inherently faulty about it.

## B. Attacks on the Collapsed Schedule

**1. "It is a Hypothetical Schedule of What Could Have Been" (Slide 26)**

This is exactly what it is supposed to be because of the specific demands of the satisfying the test of proximate causation. Of all the different types of schedules that can be generated, the only one that is not hypothetical is the As-Built schedule. If the collapsed as-built is considered hypothetical, so is the baseline or all the updates to the right side of the data date. The CAB method specifically starts with the non-hypothetical AB schedule to test the hypothesis, "The project could have been completed earlier than it actually did but for the delays at issue," and further quantifies the duration. Therefore, this criticism is really a validation of what the CAB accomplishes.

**2. "Disregards Concurrent Delays" (Slide 27)**

Because the as-built model contains all significant activities, the most prominent feature of the CAB analysis is that it forces the Contractor to squarely deal with the issue of concurrent delays

2 For more information on the Analogous Critical Path, refer to my paper published by the Project Management Institute in its Proceedings of the 1995 Symposium. This is a complex subject matter worthy of further topical treatment of its own.

by examining all delays. Properly implemented, the CAB is the method of choice for the Owner whose defense position includes the existence of concurrent delays by the Contractor. **(Slides 28-30)**

**3. "Ignores the Concept of Float"** **(Slide 31)**

It is apparent that the critic is unaware of the concept of pacing. If one party to the contract delays the project on the critical path, the delay creates additional float so that the other party to the contract can reorganize and use his or her resources more efficiently. Pacing is one party's consumption of the float created by a critical delay caused by the other party. **(Slides 32-33)**

**4. "Creates a Schedule with an Unreasonable Demand on Resources"** **(Slide 34)**

The proper execution of the CAB method includes the important process of the post-collapse review, sometimes referred to as the "Sanity Check". This process is intended to correct for logic impossibilities, weather impacts and unreasonable resource demands. The correction for the resource problem is usually done by inserting resource logic ties in the as-built prior to the next collapse iteration, or by forward-leveling the collapsed schedule.

## C. Attacks on the Process **(Slide 35)**

**1. "Susceptible to Forced Conclusions & Preconceived Conclusions"** **(Slide 35)**

There is no analysis technique that is immune from manipulation to support a predetermined position. The impacted as-planned, windows analysis or any of the other methods are equally susceptible to manipulation by an unscrupulous scheduler. The problem is not with the method itself but with how the analyst implements it.

In my opinion, the collapsed as-built method is more resistant to a forced support of preconceived conclusions than other methods because it demands that all delays be examined, without regard to who caused those delays. This is because a properly implemented collapsed as-built analysis uses the entire as-built schedule from which the analyst subtracts activities whose impact is at issue, regardless of the analyst's preconceptions or feelings as to whether they ultimately affected the project completion date. One of the most misunderstood concepts about the CAB is that the

determination of criticality is computed by the CPM calculation in the extraction process and not by the analyst when he or she selects the delay activities to be adjusted out in the collapse process. The least susceptible to after-the-fact manipulation is what I call the "As-Is Update Analysis". This approach uses the monthly updates, as-is, without subsequent modification to 'adjust for inaccuracies and errors'. However, because unaltered monthly updates submitted during the project are often inaccurate or incomplete, the 'As-Is Update Analysis' has only limited use as 'state-of-mind' indicators to gauge whether contractual delay notice provisions have been satisfied.

**2. "Ignores Cause & Effect Quantification of Individual Delays" (Slide 36)**

The CAB method, like any other method, requires a proximate cause quantification of each individual delay that is subtracted to create the collapsed schedule. Absent, this step, the analysis is not fit for trial-level presentation. The method allows for the identification of individual occurrence and quantification of both local and critical path delay impact of that occurrence.

**3. "Unreliable Because it Relies on As-Built Information" (Slide 37)**

It should be obvious that the actual start and actual finish should not be used blindly without verification of their accuracy against an independent as-built source like the daily logs. Without this verification, other methods such as the windows method are equally susceptible to the 'garbage-in-garbage-out' type of error. Properly verified as-built project information is far more reliable and resistant to manipulation than any other source of schedule input. There are ample legal precedents that not only allow, but also mandate the use of as-built information in delay analyses.

# III. The Right Approach (Slide 38)

As a manager supervising the analysis, the following should be included as part of your guidelines and directives to the analyst or the team implementing the analysis.

## A. Use a Qualified Scheduling Expert

First and foremost, use a qualified scheduling expert, not just a Primavera jock. The expert must be experienced in construction and the law but also be comfortable with the hands-on aspect of

modeling the CAB analysis. The expert should have previous successful testimony experience with CAB.

## B. Verify the As-Built Dates & Sequences

The starting point must be an as-built with verified dates and sequences. Don't just blindly rely on actual start and actual finish dates obtained from the final update. These dates must be verified against project documents such as daily reports, inspection tags and submittal logs. Many implementations fail right at the outset because the as-built is proven to be inaccurate. As they say, garbage-in, garbage-out.

## C. The As-Built Must be a Fully-Logiced CPM Model

Having verified the as-built dates, the schedule must be modeled so that the as-built dates are logic-driven and not just graphically represented as by the software as actuals. An as-built plot generated by a scheduling software using the as-built dates that are ***not*** logic-driven is the same as using a CAD program to draw the as-built. Until this proper as-built logic is inserted, no CPM analysis can take place in the technical sense. This is the main distinguishing characteristic that separates a proper implementation from the imposters and cheap imitations.

## D. Model All Significant Activities, Not Just the Critical Ones

One of the most common implementation errors is to use only the critical activities in modeling the as-built. The misguided analyst may have selected the set of critical activities based on the baseline critical path, or the successive critical paths created at each update. Whatever criterion used to define criticality, if non-critical activities are ignored, concurrent delays may also be ignored. The majority of criticisms are aimed at poorly implemented instances that violate this rule.

## E. Minimize Any Reference to the Baseline Schedule

One of the advantages of the collapsed as-built method is supposed to be that it does not have to rely on the correctness or the reasonableness of the baseline schedule. In fact, it doesn't need a baseline at all. This is because the starting point is the as-built. Therefore do not jeopardize this advantage by using the as-planned activity durations and activity relationships to collapse back to.

If you must, perform independent verification of the reasonableness of planned durations or planned logic.

### F. Disclose and Justify all Adjustments

Each and every change made to the as-built in order to create the collapsed as-built should be described and properly documented as to entitlement, causation and local delay impact. Identification of all the changes has become simple with the availability of several brands of version variance software. Discovery of undisclosed adjustments may later adversely impact the credibility of the analysis even if such failure to disclose may have been inadvertent.

### G. Verify the Buildability of the Collapsed As-Built

After the collapse process, the resulting collapsed as-built must be reviewed for reasonableness. The review should cover construction logic, resource usage, seasonal constraints and contract compliance.

## IV. Hands-On Implementation Tips

Described below are some technical tips that would help the hand-on practitioner in avoiding dead-ends and time-consuming rework.

### A. Maintain a Running Doc of Dates, Durations & Logic **(Slides 39-40)**

In my implementation using Primavera, I use the log fields to annotate each activity with the verified actual start and finish dates with the source of verification. If the activity is a delay activity, further documentary annotations should be made. I also use two of the activity code fields to carry the verified actual start and finish dates. This has proven indispensable when the logic-driven as-built model is being created.

### B. Model Seasonal Impacts Using the Calendar **(Slides 41-42)**

Instead of identifying weather and other seasonal impacts as discrete delay-activities, disable the actual impact days in the calendar. This would allow the automatic modeling of push-in (and pull-out) of work into and adverse time window for performance.

## C. Reduce Durations Rather Than Delete **(Slides 43-44)**

Reduce the delay-activity duration to zero rather than deleting the activity. This maintains the logic path from the predecessor of the delay-activity to its successor. Otherwise, a redundant 'jump tie' would be necessary. Note that in my 1995 paper, I am collapsing the model by deleting the delay activities. I found since then that this doesn't work as well as reducing the duration. Live and learn.

## D. Stick to Finish Ties and Short Lags **(Slides 45-46)**

The method requires that the baseline logic be replaced with a set of inter-activity relationship that simulates how the project was actually built. Most of the current scheduling software use the PDM convention and allows for several types of relationships (FS, SS, FF, SF) and accepts positive and negative lag values. This means that there are different ways to simulate each as-built logic tie between two activities. I have adopted logic rules based on a hierarchy of preferred relationship types using accuracy as the overriding criterion. While a full presentation of the rules require another paper, suffice to say that the rule can be summarized as follows:

- The best type of relationship type is FS with a short lag value, preferably zero.
- Whether a lag value is long or short is determined by the duration of the activities tied together by the relationship bearing the lag value.
- If the FS does not work, look for other types using the shortest lag value as the criterion.
- The FF type would be the next preferred relationship, followed by the SS.
- The SF type would be for specialized uses involving the use of Start Milestones.

In some cases, it is better to break up activities into segments in order to achieve the above and increase the accuracy and flexibility of the model.

## E. Do not Make Logic or Constraint Changes When Collapsing **(Slides 47-49)**

Rather than reducing a relationship lag value to simulate the collapse, convert that lag value into a delay-activity in the as-built model so that it can be properly labeled and collapsed as a delay

activity. Similarly, start or finish constraints should also be converted into a collapsible delay-activity rather than removing the constraint in the collapse process.

## F. Expect Numerous Iterations of the Collapse **(Slide 50)**

The post-collapse 'Sanity Check' referred to in III-G above typically results in redoing the collapse numerous times. In making the corrections, the fix must be made in the as-built model first, and then recollapsed, rather than making the correction direction in the collapsed as-built.

## G. Use one file and toggle back and forth between the two states **(Slide 51)**

Because of these multiple iterations and the requirement that all changes between the as-built and the collapse must be identified and documented, as the model becomes complex, the effort to keep the two states (as-built and the collapse) conformed and in synch become a big headache. For Primavera users, I found it helpful to create one data file that allows me to toggle between the two states using a combination of custom data fields to store the durations for the two states and using global change to replace OD and RD with those stored durations.

# V. The Easy Way – The Path of Least Resistance

No qualms about it, a correctly implemented CAB is tedious and expensive. So it would make sense not only to protect the investment by assuring correct implementation but also to soften adversarial resistance to the analysis. **(Slide 52)**

## A. Obtain a Buy-In to the As-Built Dates

Ideally, as-built dates ought to be a matter of fact, not opinion . . . at least among reasonable professionals. Try to come to some kind of agreement on the as-built dates prior to finalizing the analysis. There will be more than enough to argue about between the two parties without further clouding the issue with discrepancies on a pour date.

## B. Keep the Identity of Activities & Level of Detail Close to the Baseline

This is not because the baseline is assumed to be correct. It is to allow for a one-for-one comparison of the as-built, collapsed as-built, and the baseline schedule. Setting aside the issue of whether such comparisons are meaningful, decision makers like to compare schedules against the baseline for issues that may have nothing to do with compensable delays being addressed by the CAB.

## C. Use It In Conjunction With Other Methods

The collapsed as-built analysis is one of the most technically demanding methods. A great portion of the preparatory work can be used to generate analyses using other methodologies. For example, the tedious as-built date verification process is also necessary in order to perform the windows analysis. It would be unrealistic to expect different methods to have the same outcome with any kind of exactitude. But if the other methods produce a result that does not significantly contradict the result of the collapsed as-built method, your conclusion based on the collapsed as-built will be enhanced. Often, the collapsed as-built analysis is compared and contrasted with the windows method. When they are properly implemented, both methods produce results that are mutually consistent. Further discussion on this subject will be reserved for a separate presentation. The 1995 paper attached as an appendix contains a brief presentation on how the collapsed as-built works in conjunction with other schedules.

## D. Keep the Presentation Simple

One of the greatest reasons for resistance to a point of view supported by any analysis is lack of understanding. Especially in adversarial situations, the safest and hence the automatic reaction to something that cannot be easily understood is rejection and denial. Therefore the KISS (Keep It Simple, Stupid!) principle is mandatory to presentation. This is not to say that the underlying verification process can be ignored. For every foam-core exhibit board or a Powerpoint slide, expect to have several months of work represented by binders of back-up calculations and copies of factual source documents in addition to the usual reams of entitlement and background documentation. At the technical level CAB is a rigorous and a necessarily tedious process. The back-up documentation should mirror it but the presentation should be clear and simple.

# VI. Conclusion **(Slide 53)**

The collapsed as-built analysis is a theoretically sound method that can answer the question, 'But for the occurrences at issue when could the project have been completed?" To date, none of the criticisms and objections to the collapsed as-built method properly addresses any theoretical flaw. Rather, they ultimately rest on the issue of implementation errors. There is no method that is completely immune from dishonest manipulation or ignorant bungling. The soundness of this analysis as well as others relies on the honesty and the competence of the schedule analyst. I hope the basic guidelines discussed above will help the analyst increase his or her competence to properly implement this powerful and intuitively simple method. As for honest and responsible use of the method, that is entirely up to your integrity.

# *Collapsed As-Built*
# *The Right Way*

*Kenji P. Hoshino*

AACE International 44th Annual Meeting
June 25-28, 2000 - Calgary

# Collapsed As-Built

## Designed to Show

But for the occurrences at issue, when could the project have been completed ?

# Collapsed As-Built

## Types of Criticisms

Theoretical Flaw?

or

Implementation Error?

# Collapsed As-Built

## Subjects of Criticisms

1. Attacks on the as-built model
2. Attacks on the collapsed schedule
3. Attacks on the process

# Collapsed As-Built

## Attacks on the As-Built Model

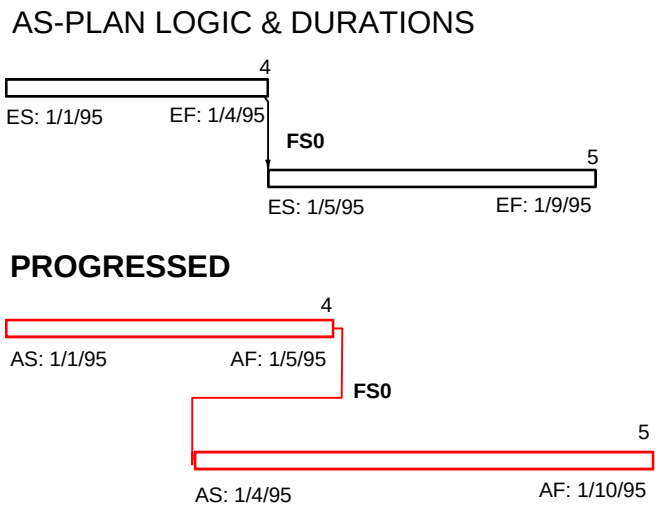
1. Ignores how the project was built
2. Assumes that the baseline schedule logic was followed
3. Assumes that no delay mitigation took place
4. Fails to identify the as-built critical path
5. Ignores the fact that the critical path is dynamic

# Collapsed As-Built

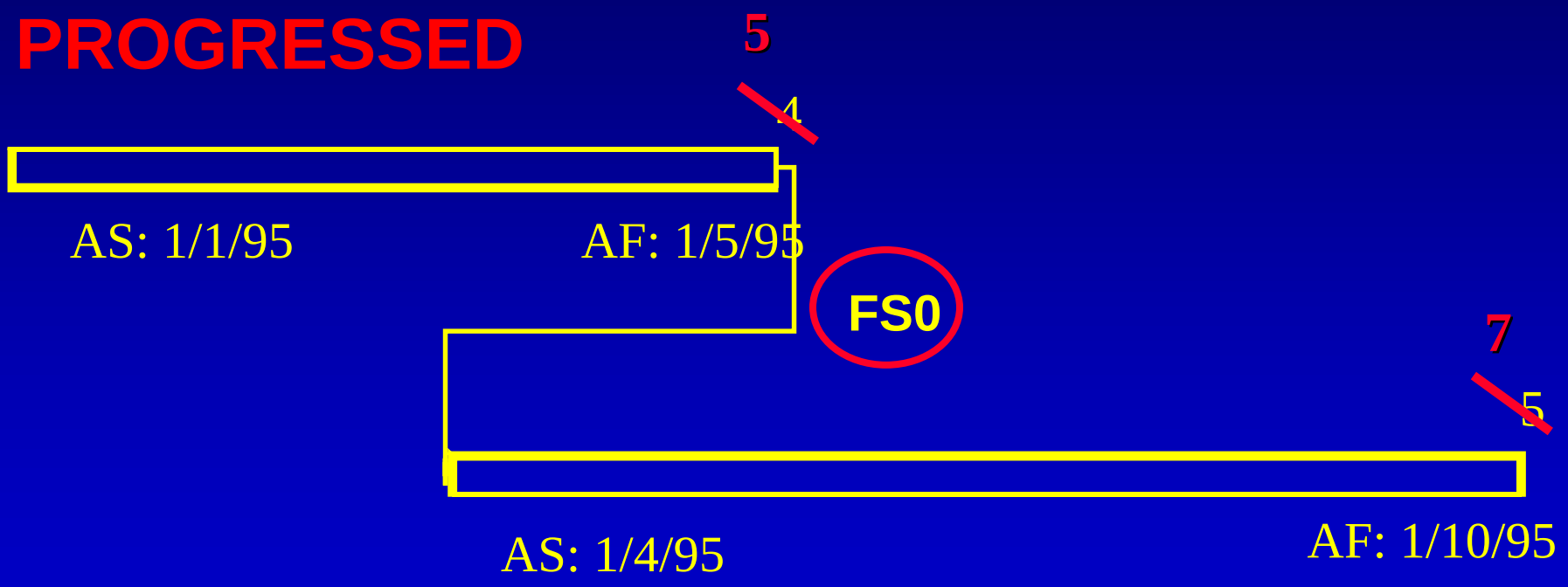
## Attacks on the As-Built Model

1. Ignores how the project was built
2. Assumes that the baseline schedule logic was followed
3. Assumes that no delay mitigation took place
4. Fails to identify the as-built critical path
5. Ignores the fact that the critical path is dynamic

# AS-BUILT LOGIC vs BASELINE LOGIC

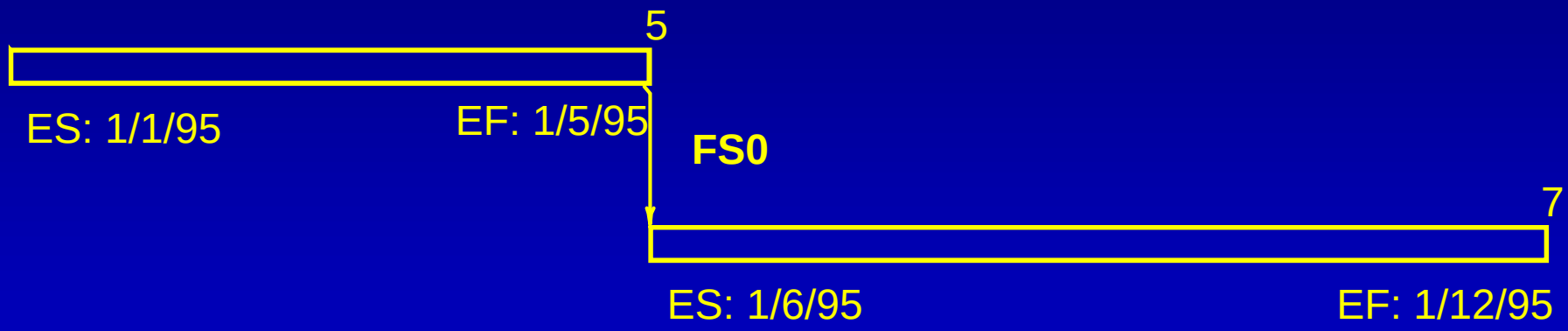


# AS-BUILT LOGIC vs BASELINE LOGIC

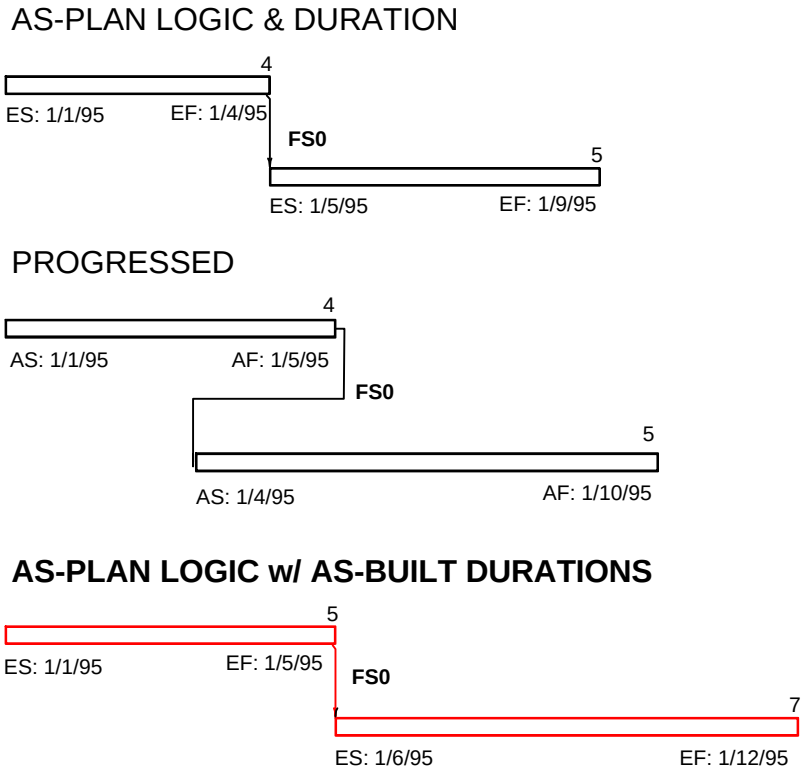


# AS-BUILT LOGIC vs BASELINE LOGIC

## AS-PLAN LOGIC w/ AS-BUILT DURATIONS

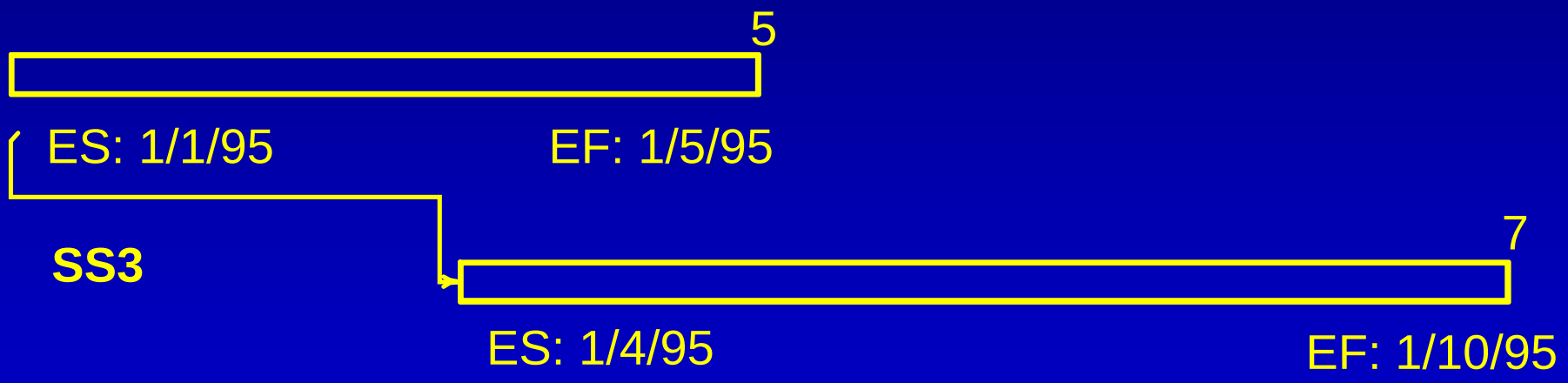


# AS-BUILT LOGIC vs BASELINE LOGIC



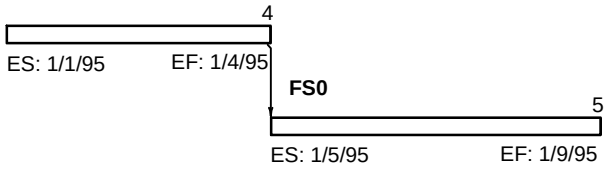
# AS-BUILT LOGIC vs BASELINE LOGIC

## AS-BUILT LOGIC w/ AS-BUILT DURATIONS

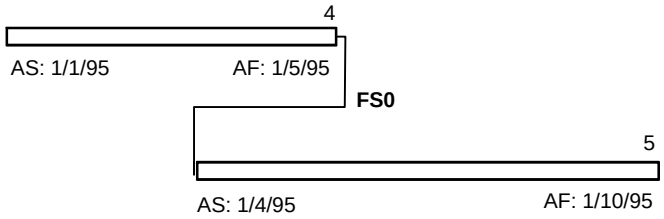


# AS-BUILT LOGIC vs BASELINE LOGIC

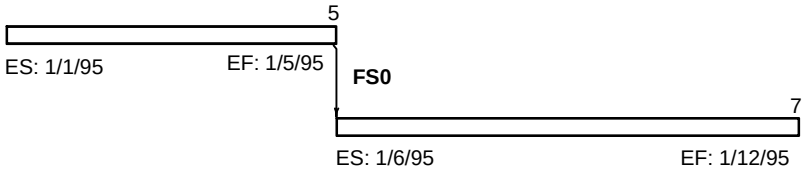
AS-PLAN LOGIC & DURATIONS



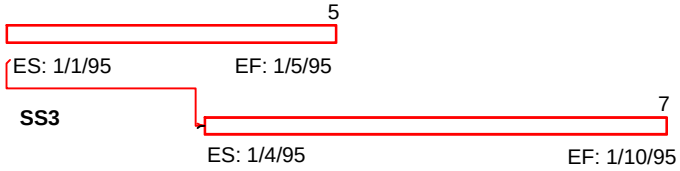

PROGRESSED




AS-PLAN LOGIC w/ AS-BUILT DURATIONS




**AS-BUILT LOGIC w/ AS-BUILT DURATIONS**



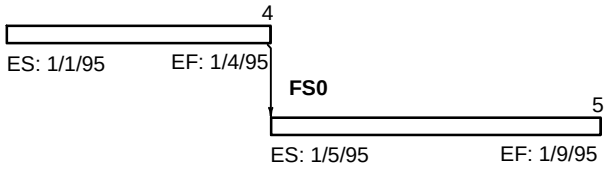
# Collapsed As-Built

## Attacks on the As-Built Model

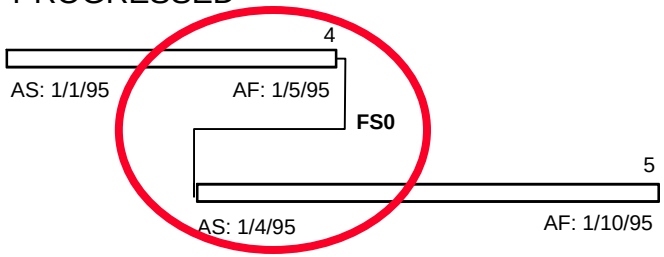
1. Ignores how the project was built
2. Assumes that the baseline schedule logic was followed
3. Assumes that no delay mitigation took place
4. Fails to identify the as-built critical path
5. Ignores the fact that the critical path is dynamic

# MITIGATION

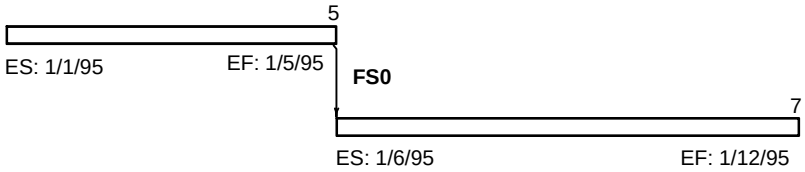
AS-PLAN LOGIC & DURATIONS



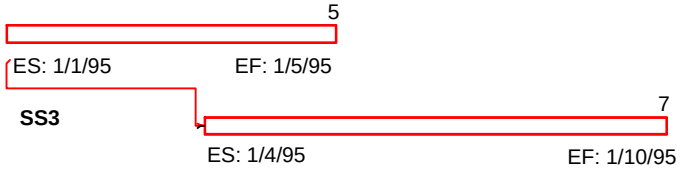

PROGRESSED




AS-PLAN LOGIC w/ AS-BUILT DURATIONS




**AS-BUILT LOGIC w/ AS-BUILT DURATIONS**



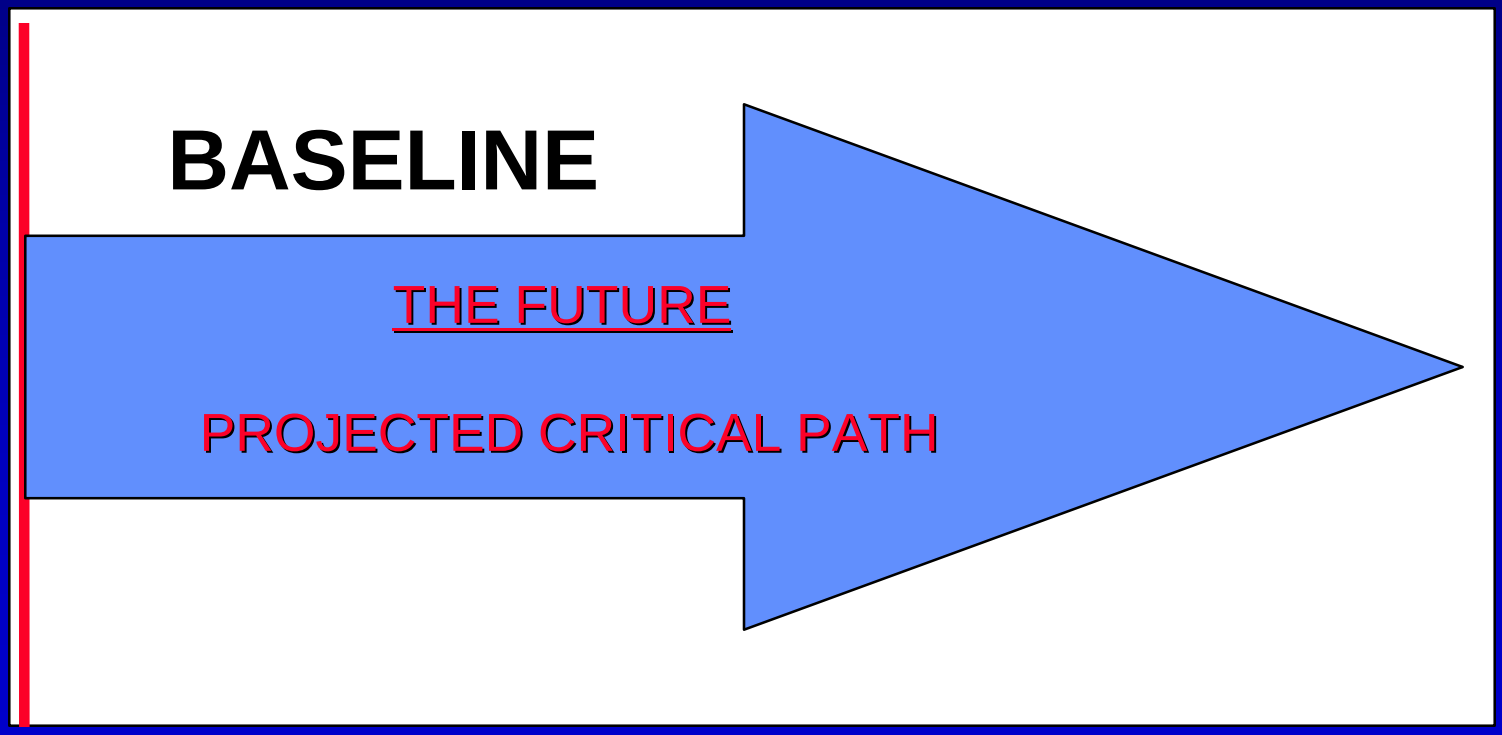
# Collapsed As-Built

## Attacks on the As-Built Model

1. Ignores how the project was built
2. Assumes that the baseline schedule logic was followed
3. Assumes that no delay mitigation took place
4. Fails to identify the as-built critical path
5. Ignores the fact that the critical path is dynamic

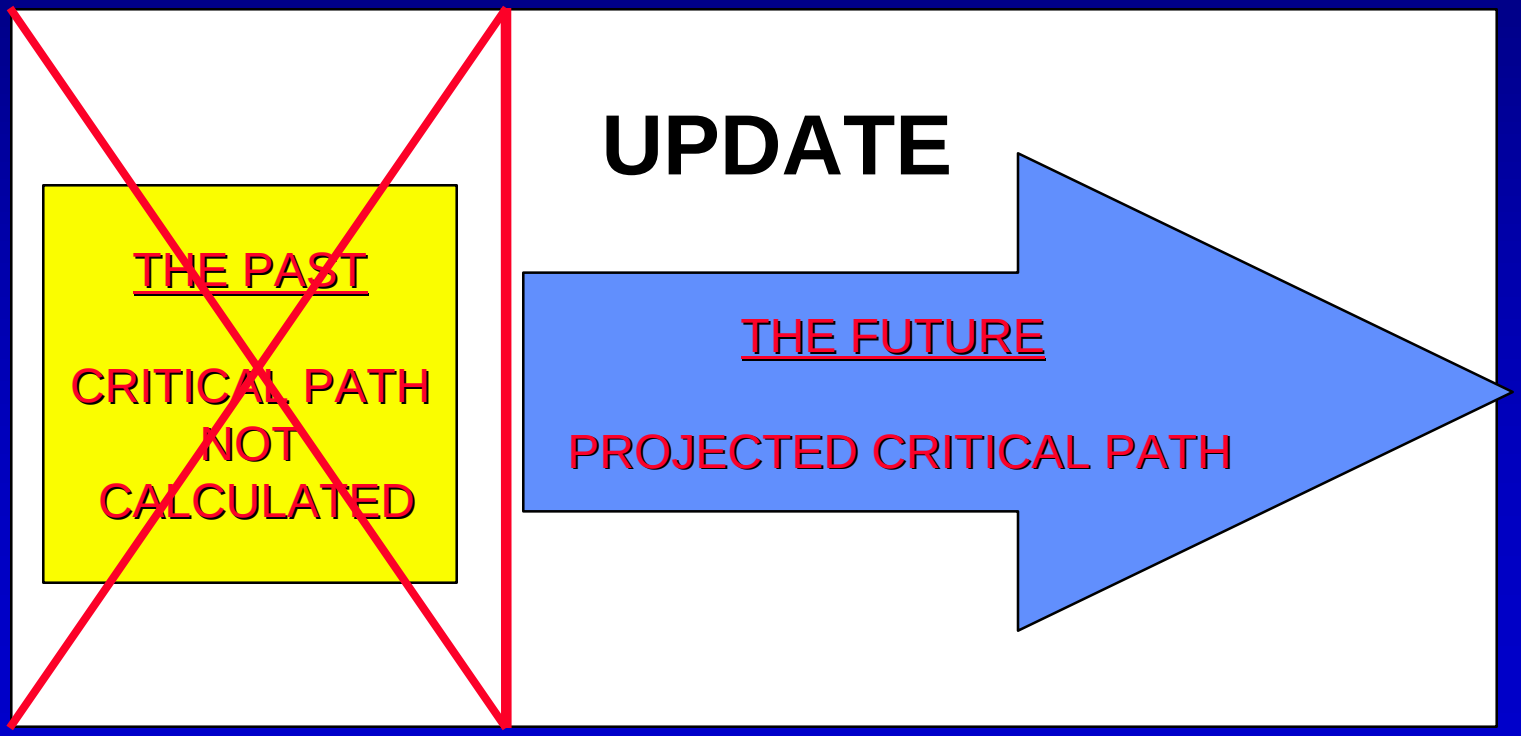
# CPM Updates
## START OF THE JOB

START FINISH

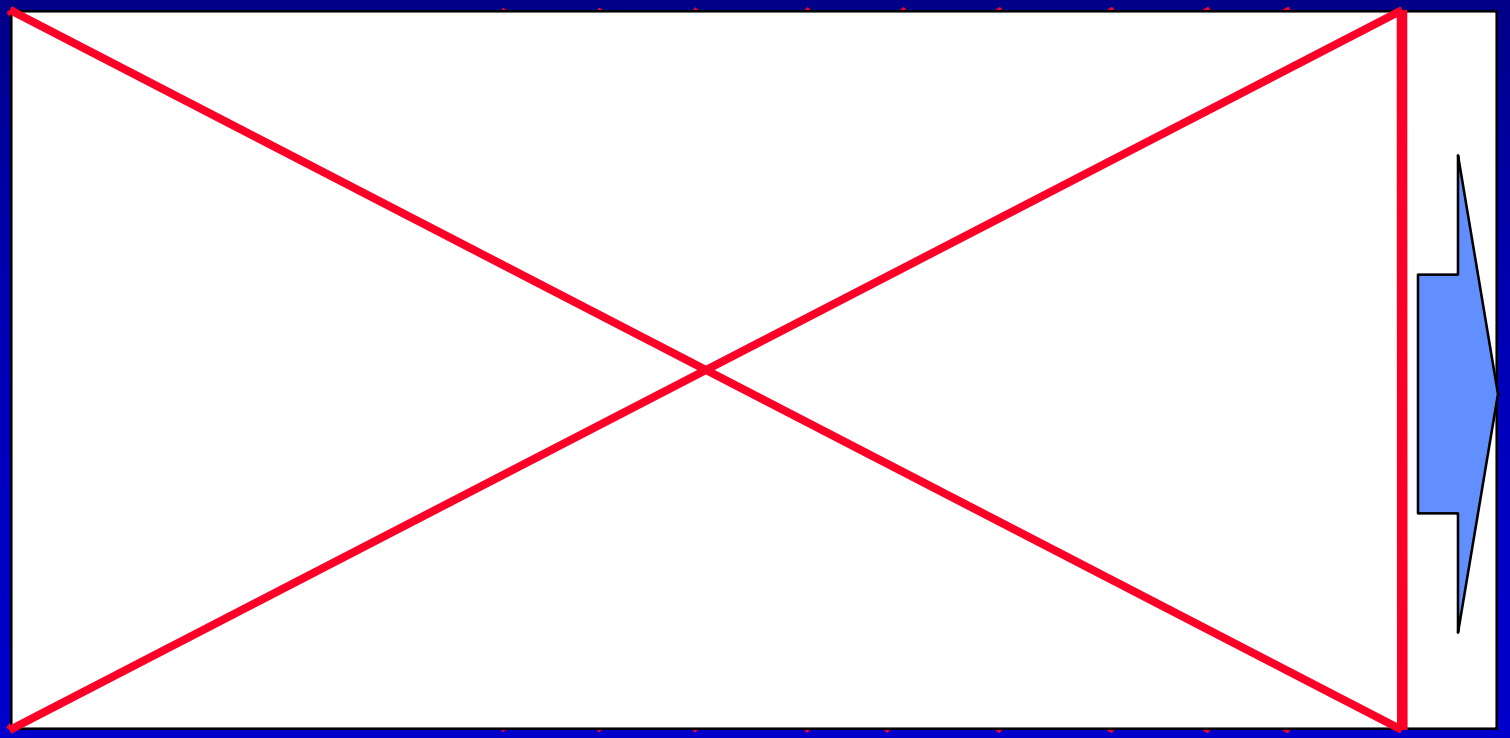


# CPM Updates
## DURING THE JOB

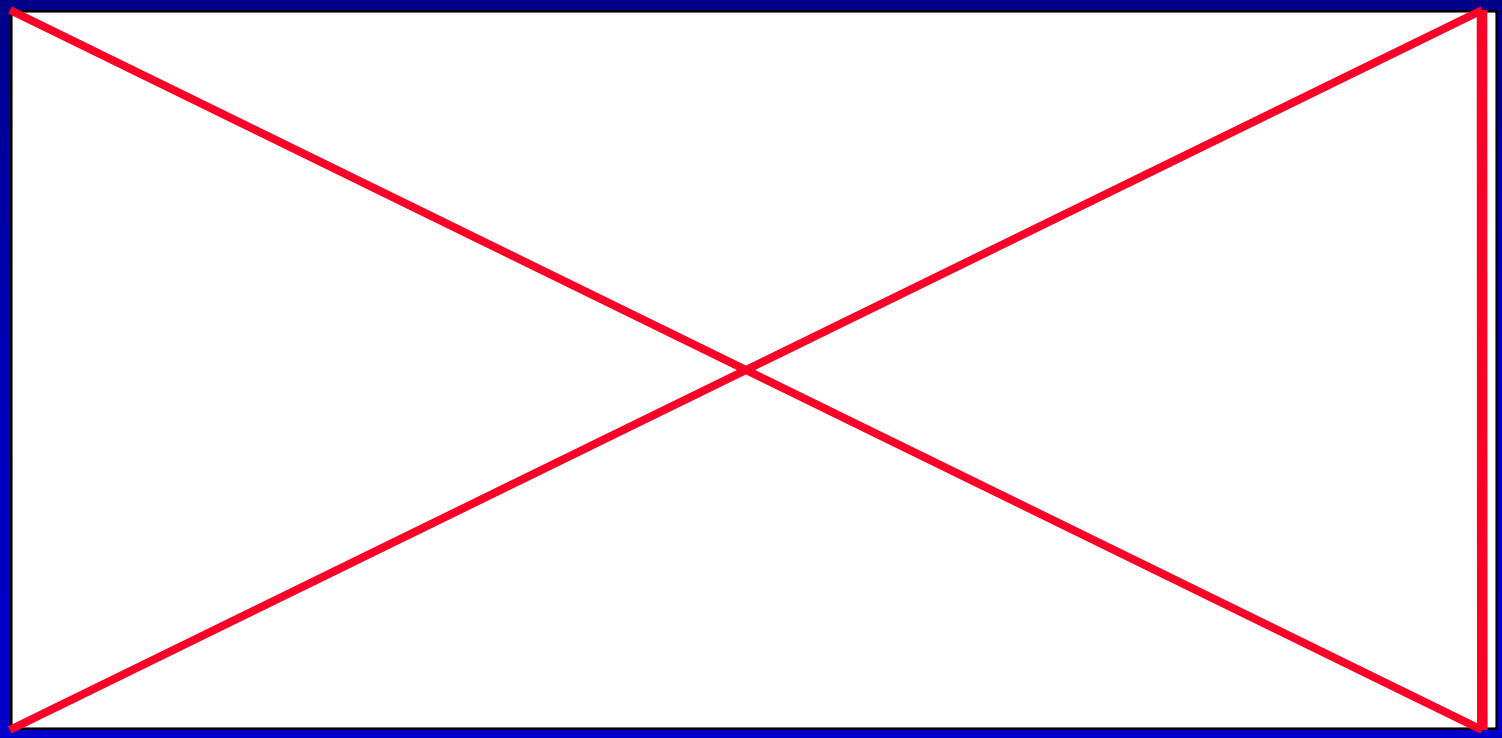
START FINISH



# CPM Updates
## DURING THE JOB



# CPM Updates
## END OF THE JOB



# CPM Updates
## END OF THE JOB

START FINISH

**FULLY PROGRESSED UPDATE**

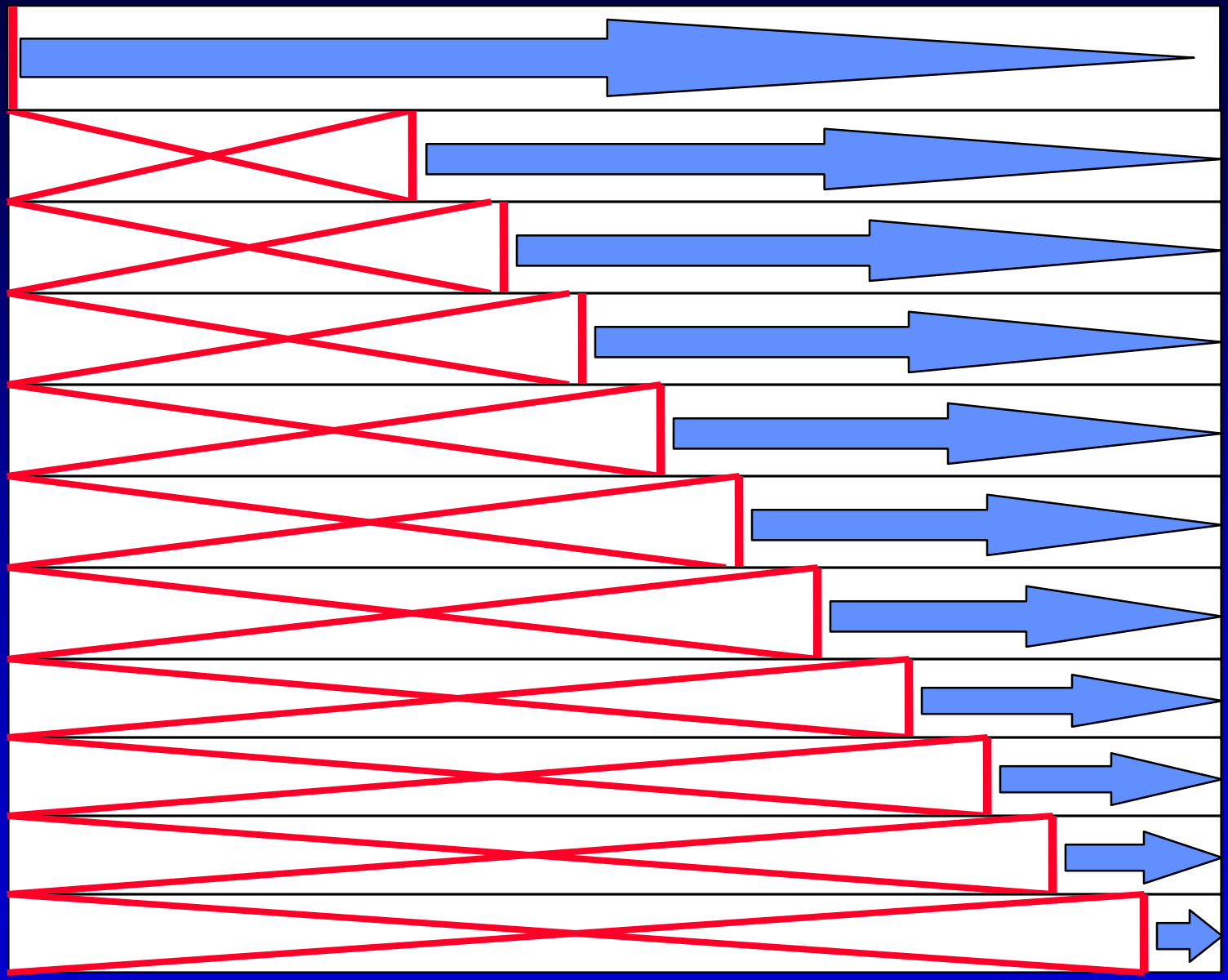
THE PAST

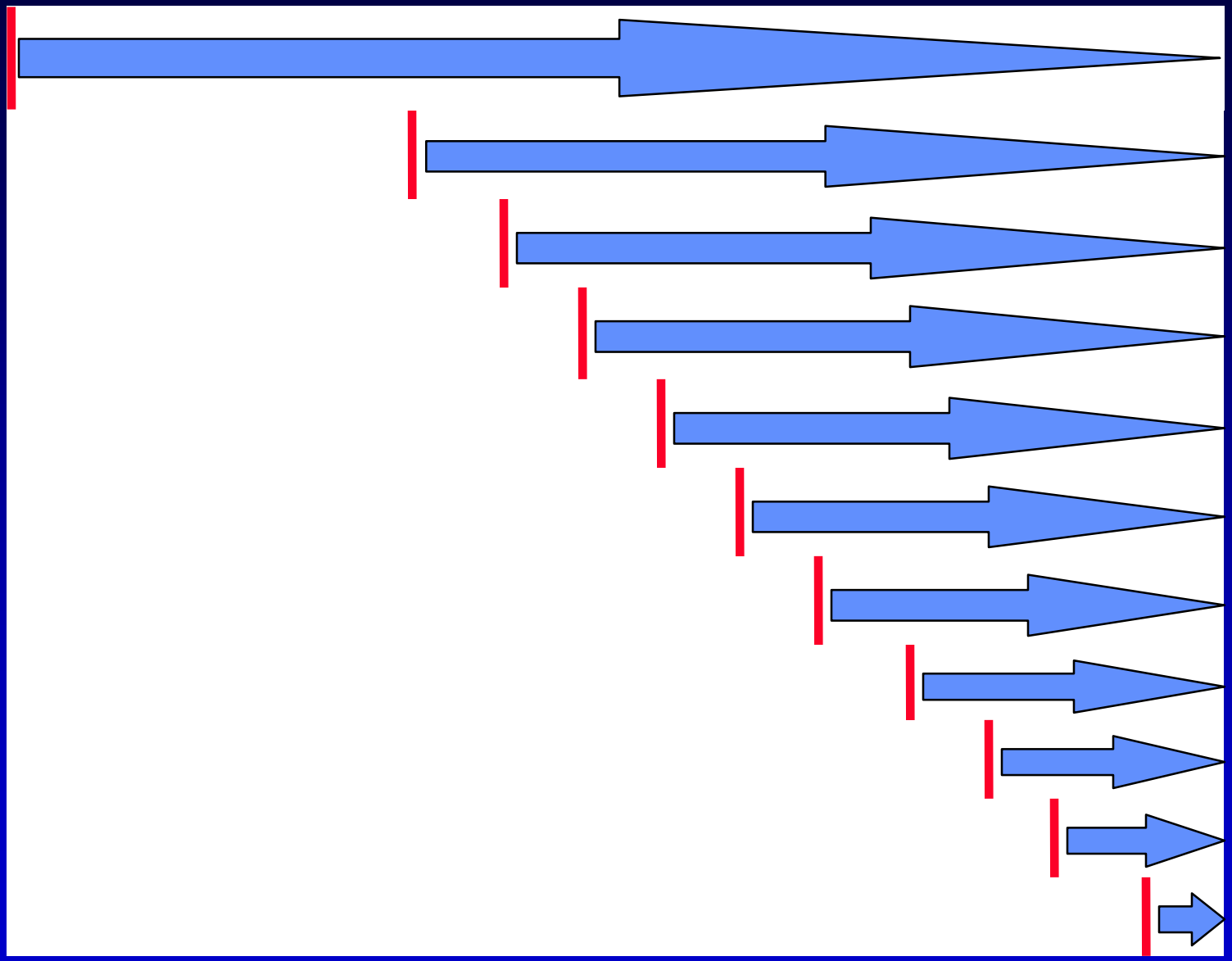
CRITICAL PATH NOT CALCULATED

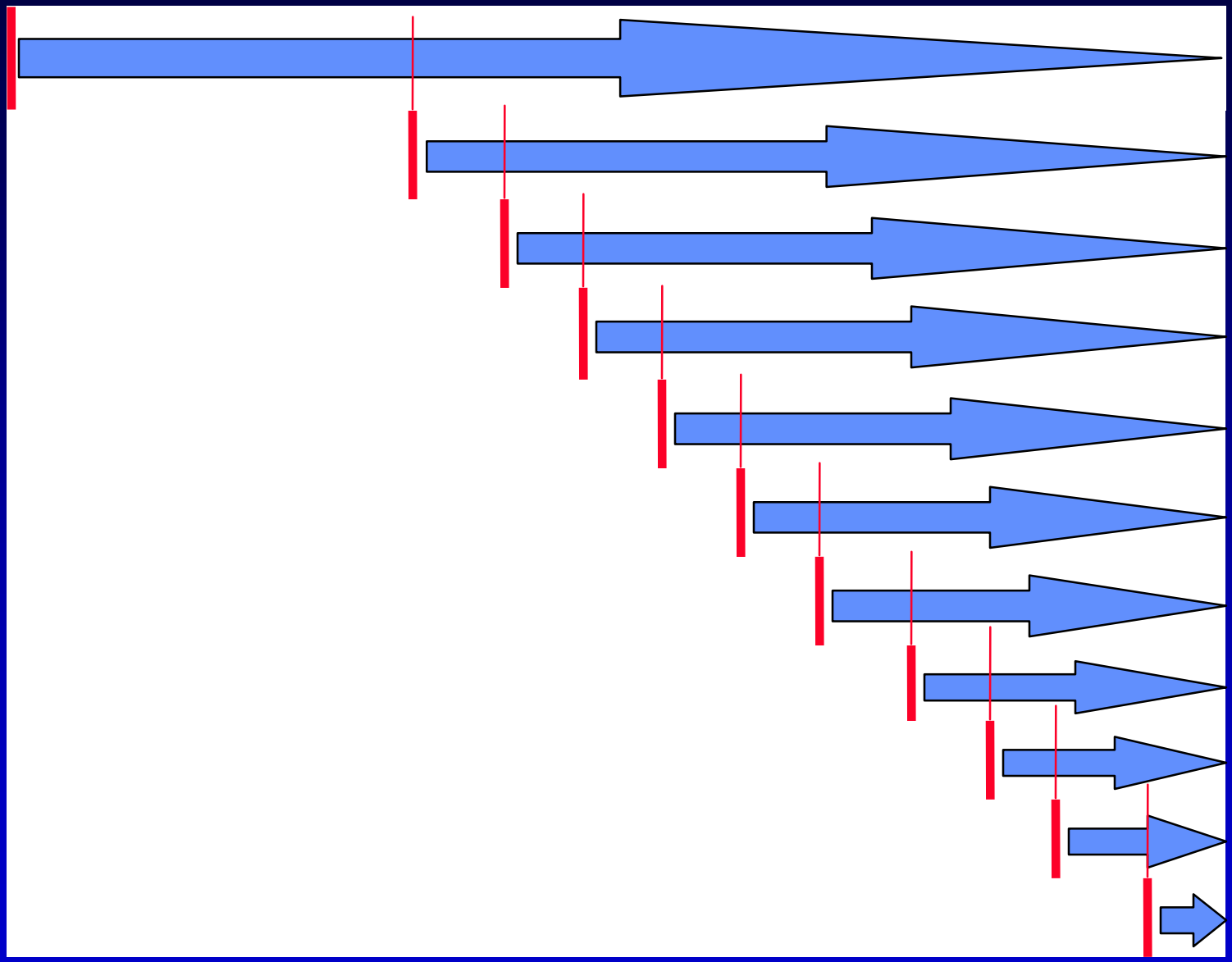
# Collapsed As-Built

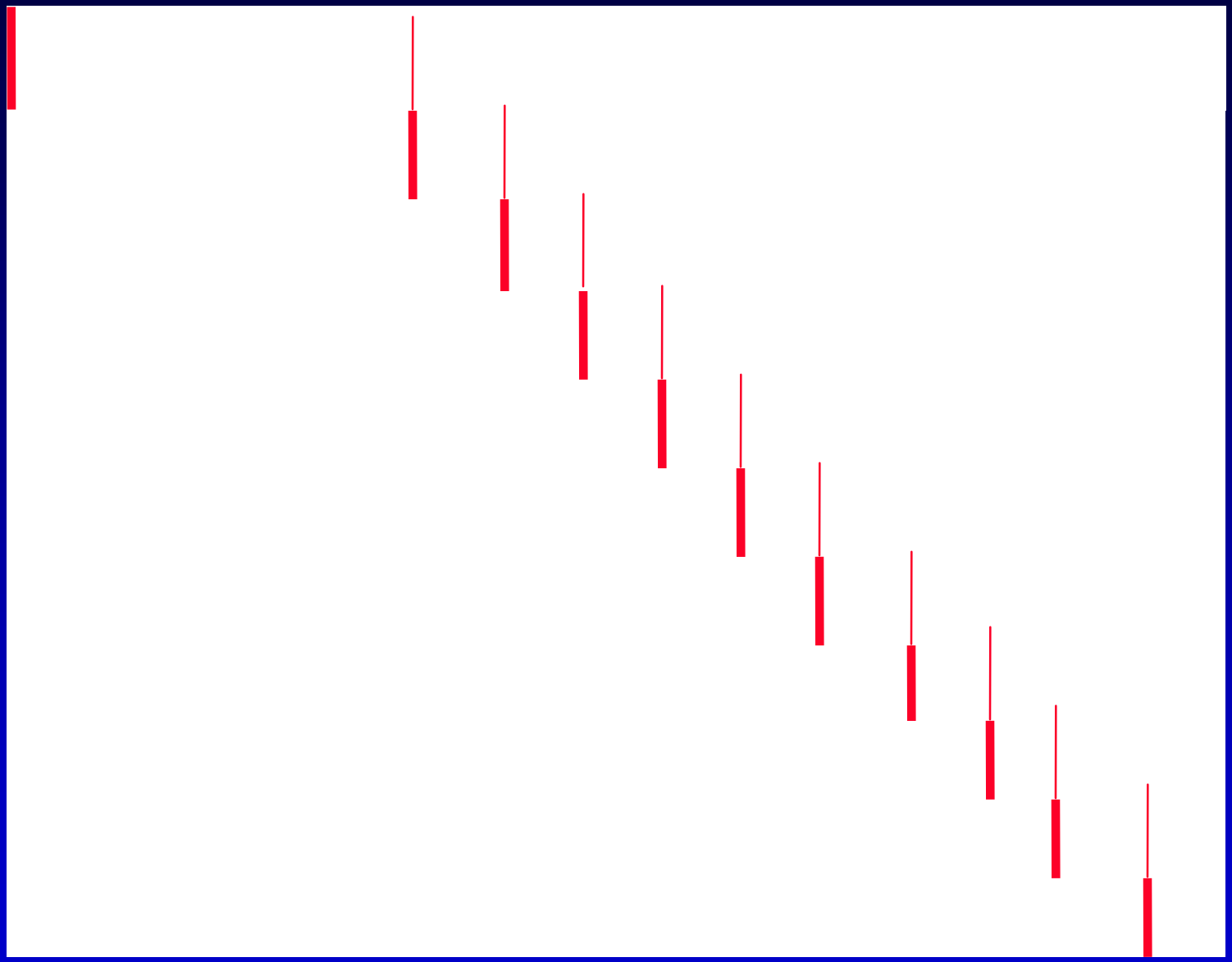
## Attacks on the As-Built Model

1. Ignores how the project was built
2. Assumes that the baseline schedule logic was followed
3. Assumes that no delay mitigation took place
4. Fails to identify the as-built critical path
5. Ignores the fact that the critical path is dynamic









# Collapsed As-Built

## Attacks on the Collapsed Schedule

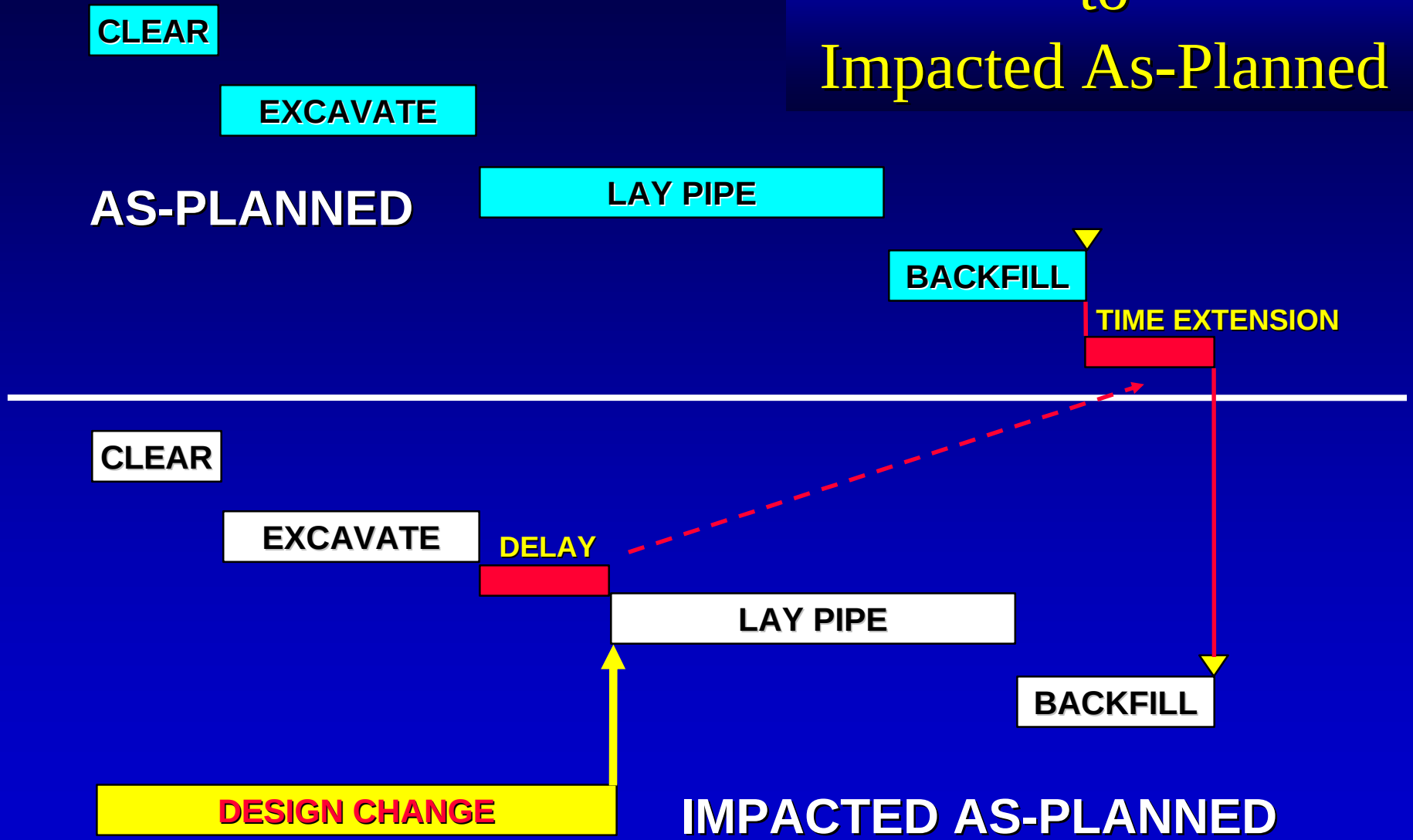
1. The Collapse is only a hypothetical schedule
2. Fails to consider concurrent delays
3. Ignores the concept of float
4. The Collapse relies on unreasonable resource loads

# Collapsed As-Built

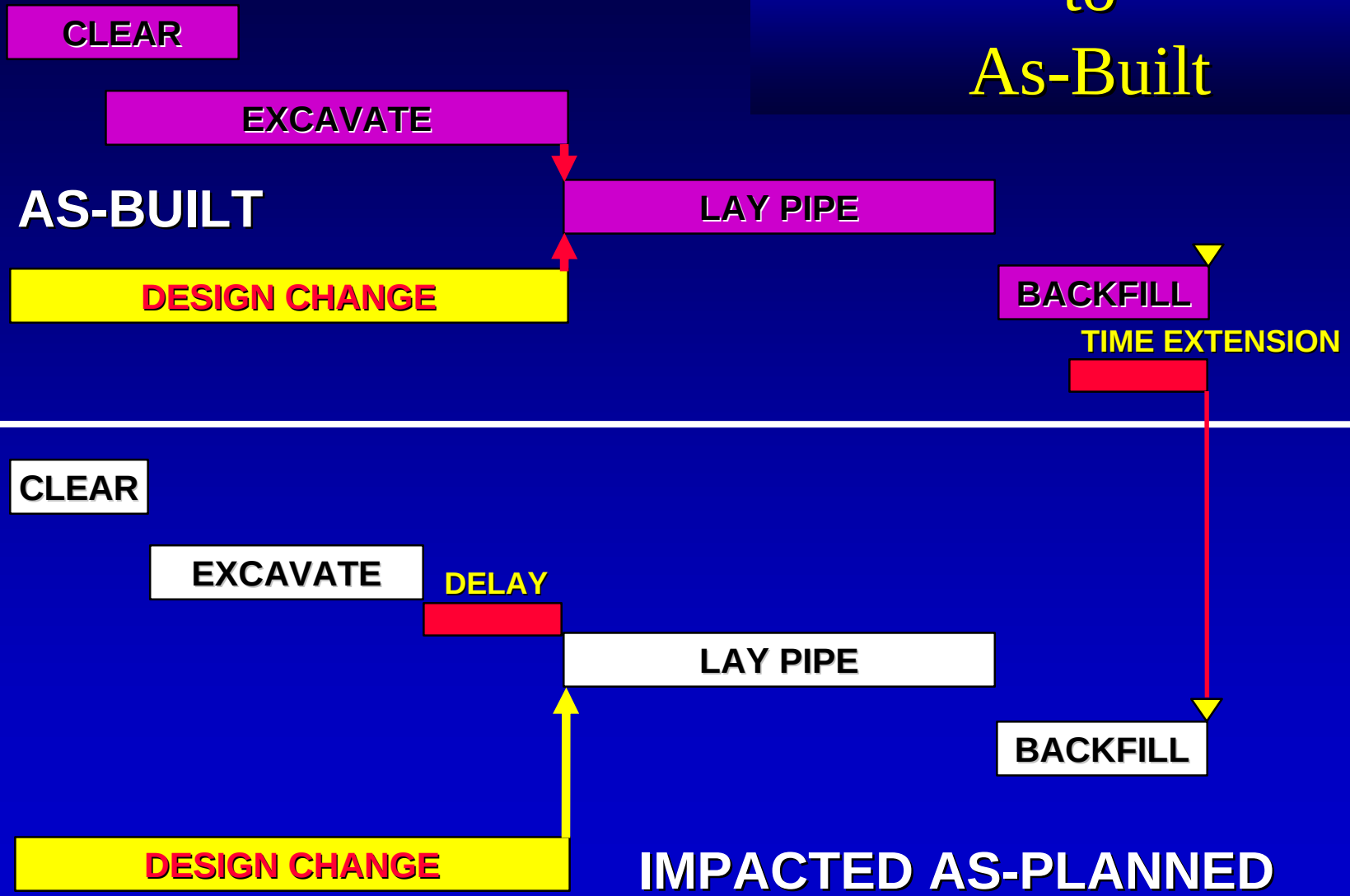
## Attacks on the Collapsed Schedule

1. The Collapse is only a hypothetical schedule
2. Fails to consider concurrent delays
3. Ignores the concept of float
4. The Collapse relies on unreasonable resource loads

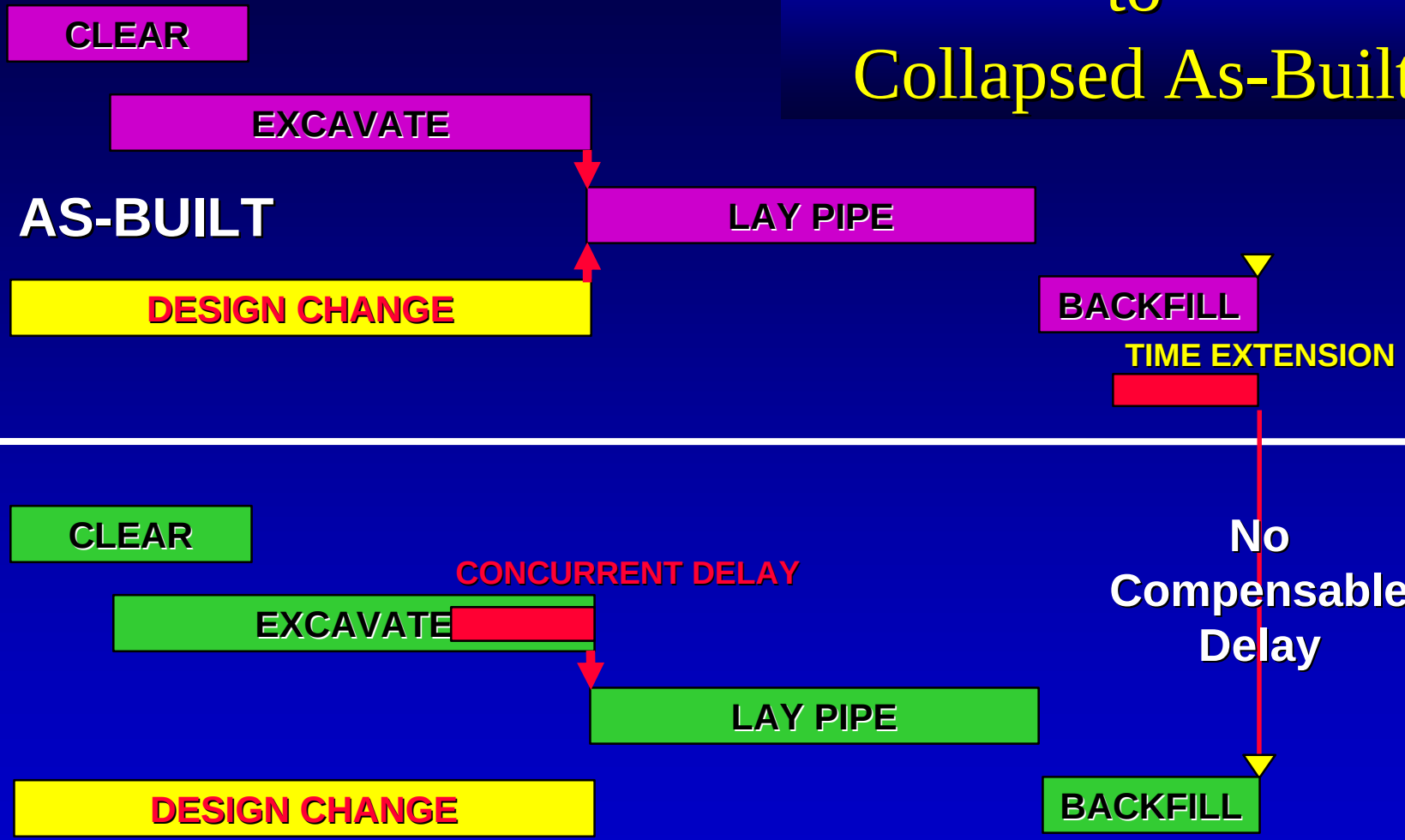
# As-Planned to Impacted As-Planned



# Impacted As-Planned to As-Built



# As-Built to Collapsed As-Built



# Collapsed As-Built

## Attacks on the Collapsed Schedule

1. The Collapse is only a hypothetical schedule
2. Fails to consider concurrent delays
3. Ignores the concept of float
4. The Collapse relies on unreasonable resource loads

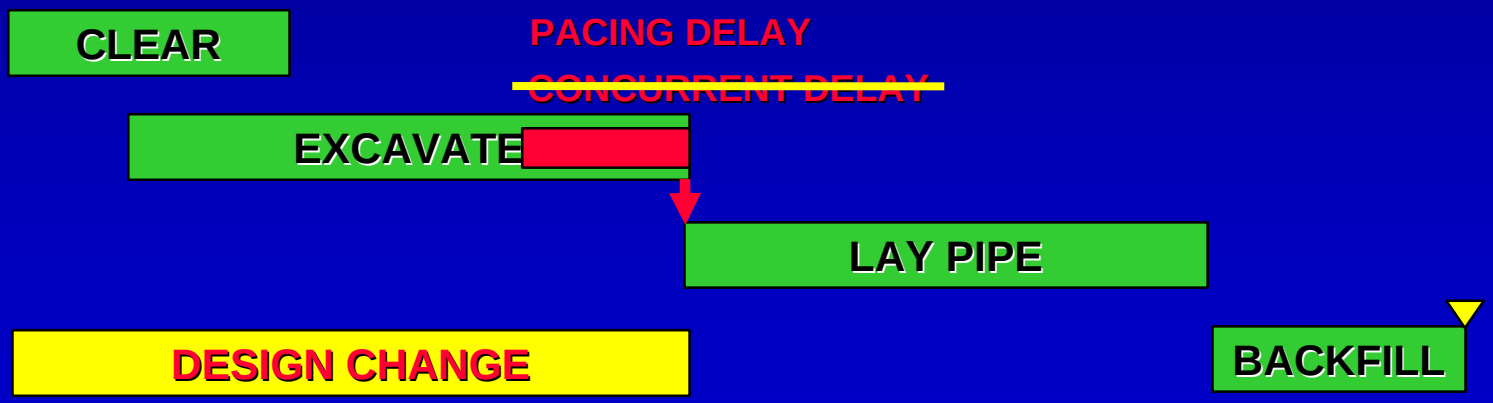
# Concurrent Delay

Where another activity independent of the subject delay is affecting the ultimate completion of the chain of activities.

# Pacing Delay

If the delay in the independent activity is the result of a conscious and contemporaneous decision to pace progress against the subject delay.

---



# Elements of Pacing

- Parent delay (subject delay)
- Conscious & deliberate decision to pace
- Contemporaneous decision to pace
  - Notice
- Present ability to resume normal progress
  - "Can finish any time"
- End date of parent delay can be determined.

# Collapsed As-Built

## Attacks on the Collapsed Schedule

1. The Collapse is only a hypothetical schedule
2. Fails to consider concurrent delays
3. Ignores the concept of float
4. The Collapse relies on unreasonable resource loads

# Collapsed As-Built

## Attacks on the Process

1. Susceptible to forced conclusions
2. Ignores causation analysis of individual delays
3. Unreliable because it relies on as-built information

# Collapsed As-Built

## Attacks on the Process

1. Susceptible to forced conclusions
2. Ignores causation analysis of individual delays
3. Unreliable because it relies on as-built information

# Collapsed As-Built

## Attacks on the Process

1. Susceptible to forced conclusions
2. Ignores causation analysis of individual delays
3. Unreliable because it relies on as-built information

# Collapsed As-Built

## The Right Approach
## Guidelines for the Manager

1. Use a qualified scheduling expert
2. Verify as-built dates & sequences
3. As-built must be a logic-driven CPM model
4. Model all significant activities, not just the critical
5. Minimize references to the baseline
6. Disclose and justify all adjustments
7. Verify the buildability of the collapsed as-built

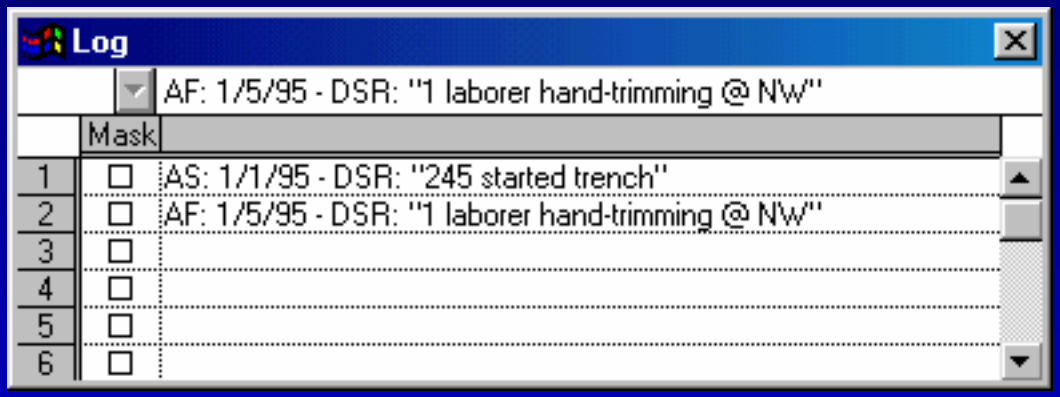
# Collapsed As-Built

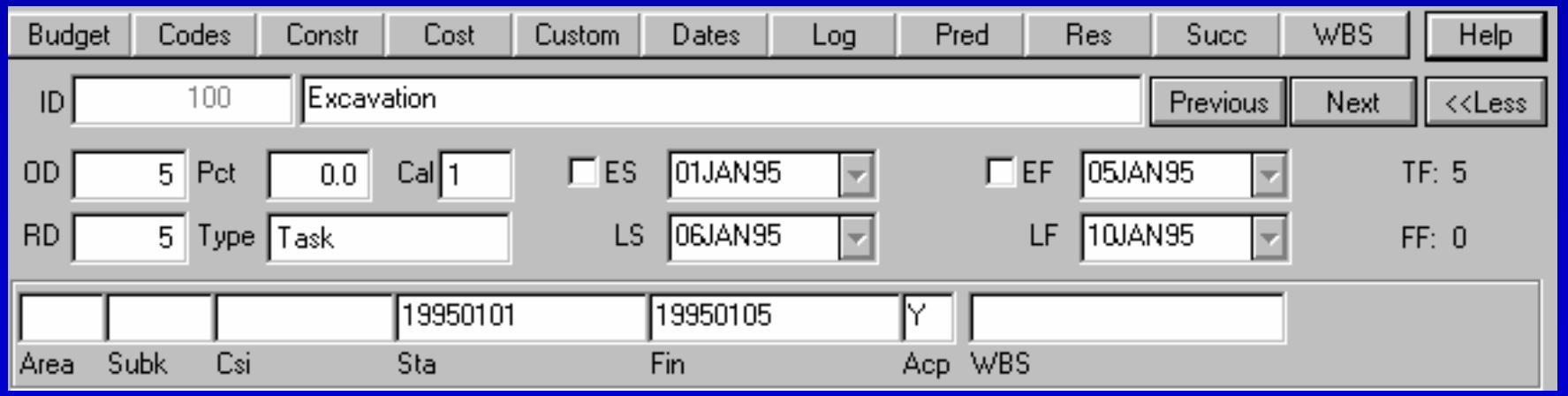
## Implementation Tips
## For the Schedule Tech

1. Document each activity as you go

# Collapsed As-Built

## Implementation Tips
## For the Schedule Tech



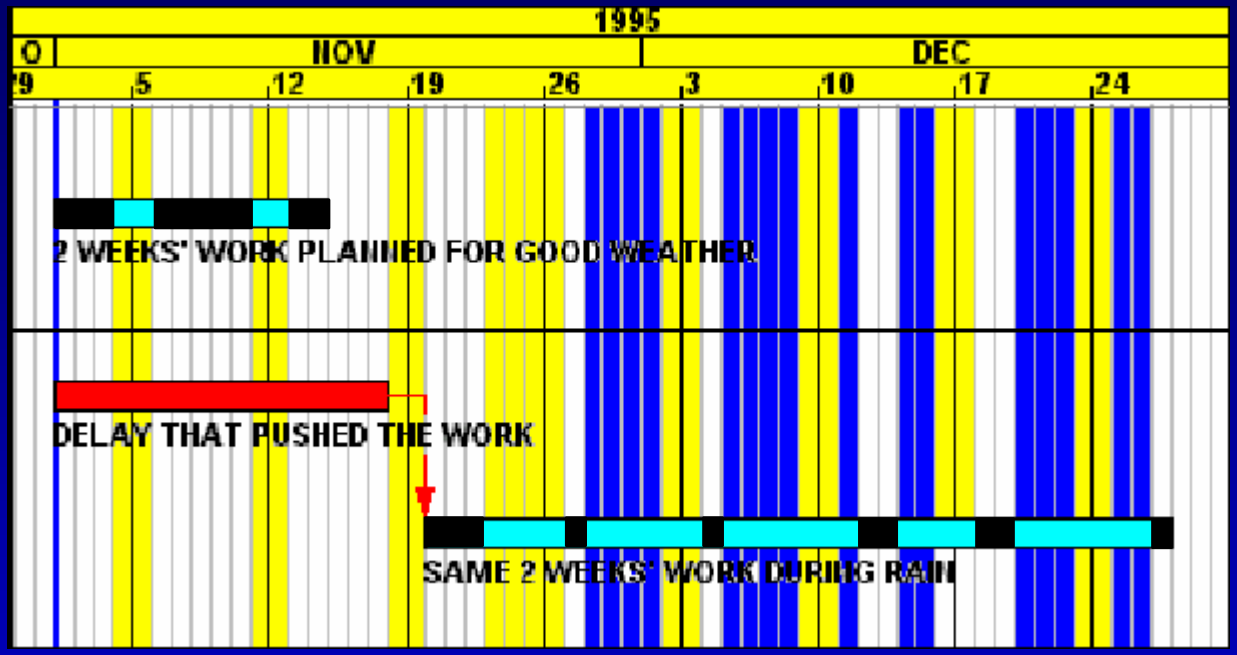
# Collapsed As-Built

## Implementation Tips
## For the Schedule Tech

1. Document each activity as you go
2. Use the calendar for seasonal delays

# Calendar-Based Modeling

3
5
2
10 wd




3
1
1
2
2
1
10 wd

November 1995

| SUN | MON | TUE | WED | THU | FRI | SAT |
|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | | |

December 1995

| SUN | MON | TUE | WED | THU | FRI | SAT |
|---|---|---|---|---|---|---|
| | | | | | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | | | | | | |

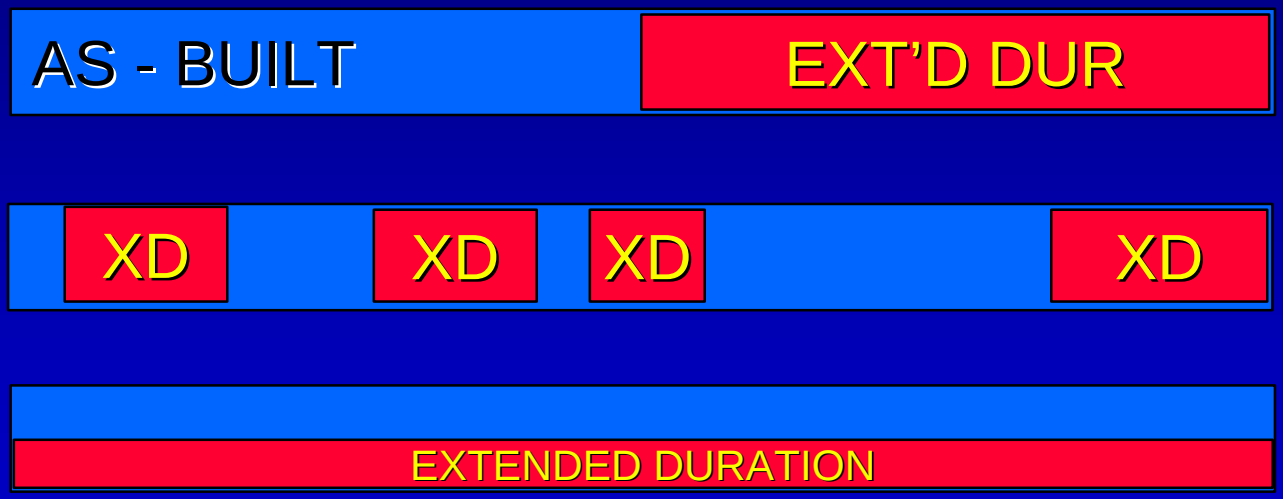
# Collapsed As-Built

## Implementation Tips
## For the Schedule Tech

1. Document each activity as you go
2. Use the calendar for seasonal delays
3. Reduce durations rather than delete the delays

# Collapsed As-Built

## Extended Duration Modeling



# Collapsed As-Built

## Implementation Tips
## For the Schedule Tech

1. Document each activity as you go
2. Use the calendar for seasonal delays
3. Reduce durations rather than delete the delays
4. Stick to finish ties and short lags

# Collapsed As-Built

## As-Built Logic Conventions

- The best type is FS with a short lag value, preferably zero
- Whether a lag value is long or short is determined by the duration of the activities tied together by the relationship
- If the FS does not work, look for other types using the shortest lag value as the criterion
- The FF type would be the next preferred relationship, followed by the SS

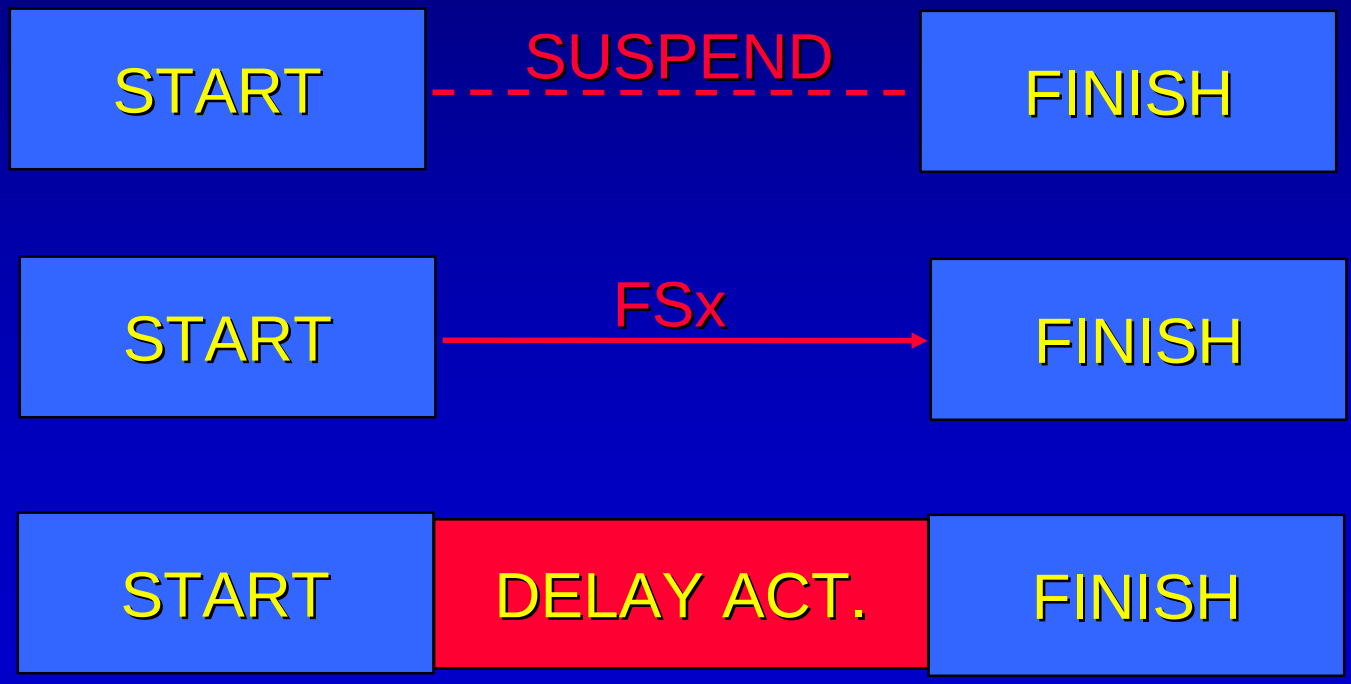
# Collapsed As-Built

## Implementation Tips
## For the Schedule Tech

1. Document each activity as you go
2. Use the calendar for seasonal delays
3. Reduce durations rather than delete the delays
4. Stick to finish ties and short lags
5. Don’t collapse logic and constraints

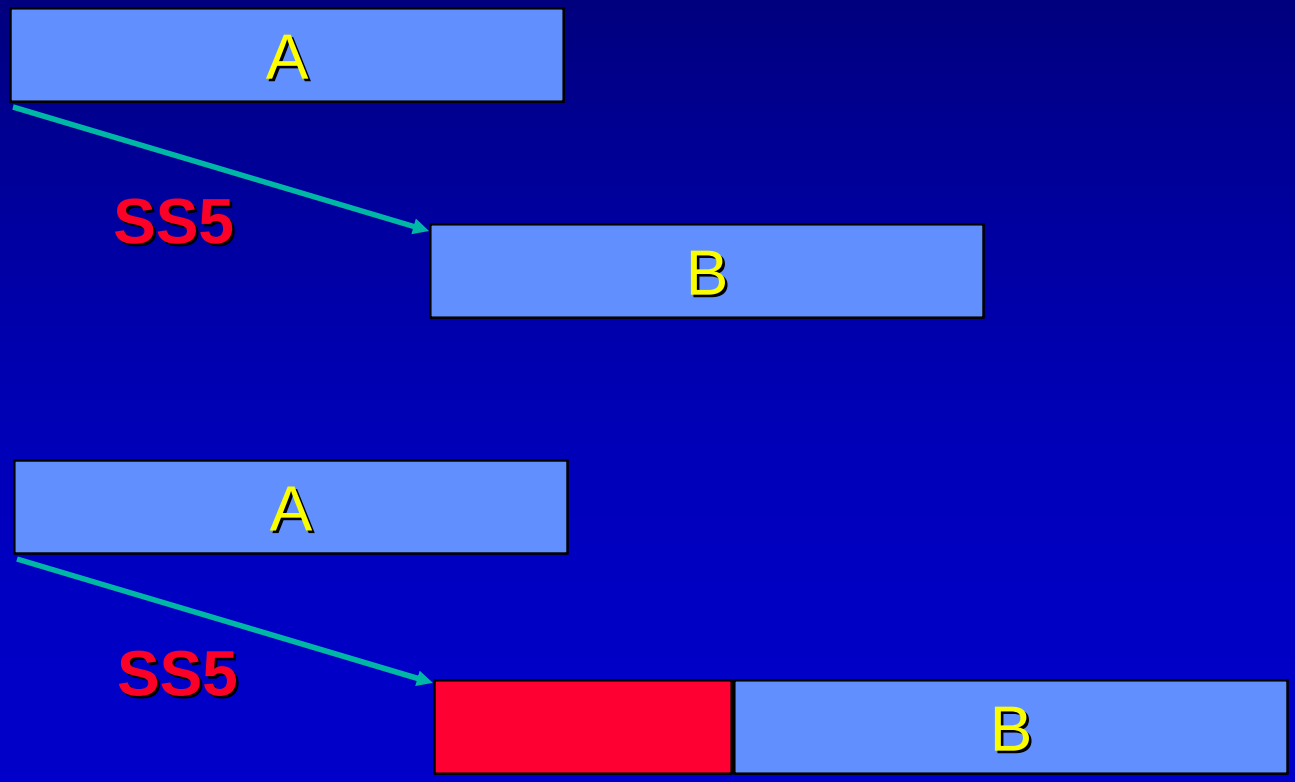
# Collapsed As-Built

## Work Suspension Modeling



# Collapsed As-Built

## Delayed Start Modeling



# Collapsed As-Built

## Implementation Tips
## For the Schedule Tech

1. Document each activity as you go
2. Use the calendar for seasonal delays
3. Reduce durations rather than delete the delays
4. Stick to finish ties and short lags
5. Don’t collapse logic and constraints
6. Expect many iterations of the collapse

# Collapsed As-Built

## Implementation Tips
## For the Schedule Tech

1. Document each activity as you go
2. Use the calendar for seasonal delays
3. Reduce durations rather than delete the delays
4. Stick to finish ties and short lags
5. Don’t collapse logic and constraints
6. Expect many iterations of the collapse
7. Use one toggled file during modeling

# Collapsed As-Built

## Path of Least Resistance

1. Agree on as-built dates
2. Allow for easy comparison to the baseline
3. Use it in conjunction with other methods
4. K.I.S.S. the presentation

# Collapsed As-Built

## Conclusion

~~Theoretical Flaw?~~

or

Implementation Error?

# Collapsed As-Built

## The Right Way

Kenji P. Hoshino
(702) 807-3576