فرآیند جامع آنالیز تاخیرات در پروژه

بخش نهم:

آنالیز زمانی تاخیرات

آنالیز زمانی تاخیرات بعد از وقوع تاخیر و اختلاف
(After-the-Fact Schedule Analysis)

ارائه دهنده:

سید محمدرضا علوی‌پور

CMAA Authorized Trainer

(PhD, PMP, CMIT)

دانشگاه Illinois Tech آمریکا

What we will learn?

- After-the-Fact schedule analysis
  - Concepts
  - Retrospective IAP and TIA
  - Time slice windows analysis (TSWA)
  - As-planned vs. as-built windows analysis (APAB)
  - Retrospective longest path analysis (RLPA)
  - Collapsed as-built analysis (CAB)
- How to choose the methods?
- The most three types of logic
- Effects of three logic types on delay analysis

After-the-Fact Schedule Analysis

(Concepts)

Why after-the-Fact schedule analysis needed?

- **Contractors** **frequently** **assert** that they have been **delayed** for **reasons** **beyond** their **control**.
- **Owners** **often remain** **unconvinced** that the Contractor is legitimately entitled to a **time extension or delay**, **acceleration** and **loss of productivity damages**.
- **Large dollar amounts** may **hinge** upon the **outcome of a dispute** over **project delay**.

Consequently, a thorough retrospective schedule analysis of all project delays is essential for the equitable resolution of delay and impact-related construction disputes.

Why after-the-Fact schedule analysis needed?

- **Owner** may be **contractually** entitled to **recover** **either**:
    1. **Liquidated damages** for delay caused by the contractor
    2. **Actual damages** for delay caused by the contractor

Observational vs. Modeled

- **Forensic schedule analyses** generally can be **classified** into **two types of methodologies**:

1) **Observational**

   - Observational techniques typically **consist** of using (1) **contemporaneous schedules** + (2) **only modifications** for **obvious** and **significant errors**.

2) **Modeled**

   - Modeled techniques **include** **(1) addition** or **subtraction** of **activities** **and or** (2) **logic** to **model impact events**.

Do we need to change and correct the schedule even after-the-fact?

Why after-the-Fact schedule analysis needed?

- The **Prospective methods** of delay analysis may **no longer** be **appropriate** if the **delay** already **occurred**.
  - **CP 4** should **not be used** in this case
    - ✓ Then, there should be a **provision** that **addresses** the consideration of EOT applications:
      1. **After completion of works**
      2. **After the occurrence of delay events or its impact**

Contracts should include provisions as to what procedure should be used when delays already occurred.

Changing schedules after-the-fact

ASCE Standard ANSI/ASCE/CI 67-17 (Schedule Delay Analysis)

- **Courts and boards** typically attach a presumption that **contemporaneous CPM** schedules are **correct** and

schedules should be **presumed correct unless otherwise** shown to be **inaccurate**.

See, e.g., *Appeal of Santa Fe, Inc.*, VABCA No. 2168, 87-3 B.C.A. (CCH) ¶ 20104 (Aug. 25, 1987) "We do not find that the November revisions were necessary to "correct" the October CPM. Rather, the November CPM merely reflected a different plan for further prosecution of the work which differed from the previous plan. However, the adoption of an alternate method of performance in the later CPM does not, of itself, contradict the existence of a delay as shown in the preceding CPM. There is a rebuttable presumption of correctness attached to CPM's upon which the parties have previously mutually agreed. In the absence of probative evidence, not present in this appeal, that the delay shown was not in fact sustained we will rely on the October CPM for the period it was in effect, i.e., through the end of November 1981. For the subsequent period, we will rely on the November CPM. To put it another way, in the absence of compelling evidence of actual errors in the CPM's, we will let the parties 'live or die' by the CPM applicable to the relevant time frames."

Changing schedules after-the-fact

ASCE Standard ANSI/ASCE/CI 67-17 (Schedule Delay Analysis)

- **Changes** to schedules made after the fact, such as **after project completion**, should be **limited to special circumstances**.
  - Changes to the schedules may include changing
    1. Activity durations,
    2. Logic,
    3. As-built dates,
    4. Constraints, or
    5. Other schedule information often kept in the schedule software.

When changes may be appropriate?

Changing schedules after-the-fact

ASCE Standard ANSI/ASCE/CI 67-17 (Schedule Delay Analysis)

- When **significant errors** are found in the schedule, **after-the-fact schedule** adjustment to fix the errors **may be allowable**.

See, e.g., *Neal & Company, Inc. v. U.S.*, 36 Fed. Cl. 600 (1996) analysis accepted where government's expert prepared a new baseline schedule after the project to add in logic ties for labor that were missing, then re-updated the schedule.

Changing schedules after-the-fact

ASCE Standard ANSI/ASCE/CI 67-17 (Schedule Delay Analysis)

- For example, logic errors, missing scope, or palpable incorrect durations may have to be **adjusted** before utilizing the schedule for **delay analysis purposes**.

See, e.g., *Appeals of States Roofing Corp.*, ASBCA No. 54860, 10-1 B.C.A. (CCH) ¶ 34356 (Jan. 12, 2010) "In any event, the credible evidence supports the finding that the adjustments [contractor's expert] made were exactly as he described them: minor corrections to logic errors relating to submittals. The Navy did not dispute that there were such logic errors in the schedule and, to the extent there were, we consider correction of them to be in keeping with good scheduling practices.");

Which logic should be corrected?
Should we also change discretionary constraints?

Changing schedules after-the-fact

ASCE Standard ANSI/ASCE/CI 67-17 (Schedule Delay Analysis)

10.5 Changes to schedules generally may be made to correct **<u>necessary physical</u>** or **<u>contractual constraints</u>**

but typically **not the contractor's preferential sequencing**

*CPM schedules generally should be **modified** only <u>after the fact</u> to **correct errors** in <u>physical constraints</u> or <u>contractual constraints</u>*

The <u>burden of proof</u> of changing lies with whom?

Changing schedules after-the-fact

ASCE Standard ANSI/ASCE/CI 67-17 (Schedule Delay Analysis)

- **Burden of proof** of changing the schedules **after they are prepared and used during the project**

  lies with the **party making the changes**.

  As a result, any changes made after the fact should be made with good justification and be carefully documented and explained.

  See, e.g., *J. A. Jones Construction Co.*, ENGBCA No. 6252, 97-1 B.C.A. ¶ 28,918 (1997) "The more a contractor departs in litigation from its contemporaneously-prepared schedules, the greater the need to explain and justify the reasons and assumptions underlying such departures."

So, let's look at the FSA methods?

FSA methods

- There are **several industry accepted methodologies** for **forensic schedule analysis**.
- **SCL** says **four forensic** methods
- **AACE** says **nine forensic** methodologies
- **Others** say **seventeen or more**
- **An argument** could be made for a **near infinite number**
- We will discuss **major types**

In order to avoid or at least minimize disputes over methodology, it is recommended that the parties try to agree an appropriate method of delay analysis for both contemporaneous and after-the-fact.

FSA methods – SCL Protocol

Method of Analysis	Analysis Type	Critical Path Determined	Delay Impact Determined	Requires
Time Slice Windows Analysis	Effect & Cause	Contemporaneously	Retrospectively	• Logic linked baseline programme. • Update programmes or progress information with which to update the baseline programme.
As-Planned versus As-Built Windows Analysis	Effect & Cause	Contemporaneously	Retrospectively	• Baseline programme. • As-built data.
Retrospective Longest Path Analysis	Effect & Cause	Retrospectively	Retrospectively	• Baseline Programme. • As-built programme.
Collapsed As-Built Analysis	Cause & Effect	Retrospectively	Retrospectively	• Logic linked as-built programme. • A selection of delay events to be modelled.

FSA methods - AACE RP29R-03

Observational	As-Planned As-Built	Gross	As-Planned vs. As-Built (MIP 3.1)
		Periodic	As-Planned vs. As-Built (MIP 3.2)
	Contemporaneous Period Analysis (Windows)	Contemporaneous As-Is	Contemporaneous Period Analysis (MIP 3.3)
		Bifurcated Contemporaneous	Bifurcated CPA (MIP 3.4)
		Recreated / Modified	Recreated CPA (MIP 3.5)
Modeled	Time Impact Analysis	Single Base	Impacted As-Planned (MIP 3.6)
		Multiple Base	Retrospective TIA (MIP 3.7)
	Collapsed As-Buiult	Single Simulation	Collapsed As-Built (Single) (MIP 3.8)
		Multiple Simulation	Collapsed As-Built (Multiple) (MIP 3.9)

FSA methods – Comparison

The Information Required					
	Baseline programme	Logic linked baseline programme	As-built data	Delay events to model	Up to date programmes/ process information
Time Slice Windows					
As-Planned v As-Built Windows					
Longest Path					
Collapsed As-Built					

The table above shows the evidence required for each delay analysis method. The table has been simplified.

After-the-Fact Schedule Analysis

(Retrospective IAP and TIA)

Retrospective time impact analysis – Definition

- **AACE** Talks about this **method**
- **Very similar** to **prospective time impact analysis**

1. **Baseline** can be **used** for this analysis (**Impacted as-planned**)
2. **Contemporaneous update** can be **used** for this analysis (**Retrospective time impact analysis**)

- AACE **method implementation protocol (MIP) 3.7and 3.8** say:
  - ✓ **Impacted as-planned** is **less accurate** than **retrospective time impact analysis.**

Retrospective time impact analysis – How to use?

1. Take a **contemporaneous schedule and revise** it based on **actual** situation = **As-built**
2. Take a **delay event**
3. Use its **actual duration (start** and **finish** times**)**
4. **Insert actual duration** and a **delay event** into the **contemporaneous schedules** that is **revised to date = As-built**
5. **Show** that **event's alleged impact** on the **contractor's update**.

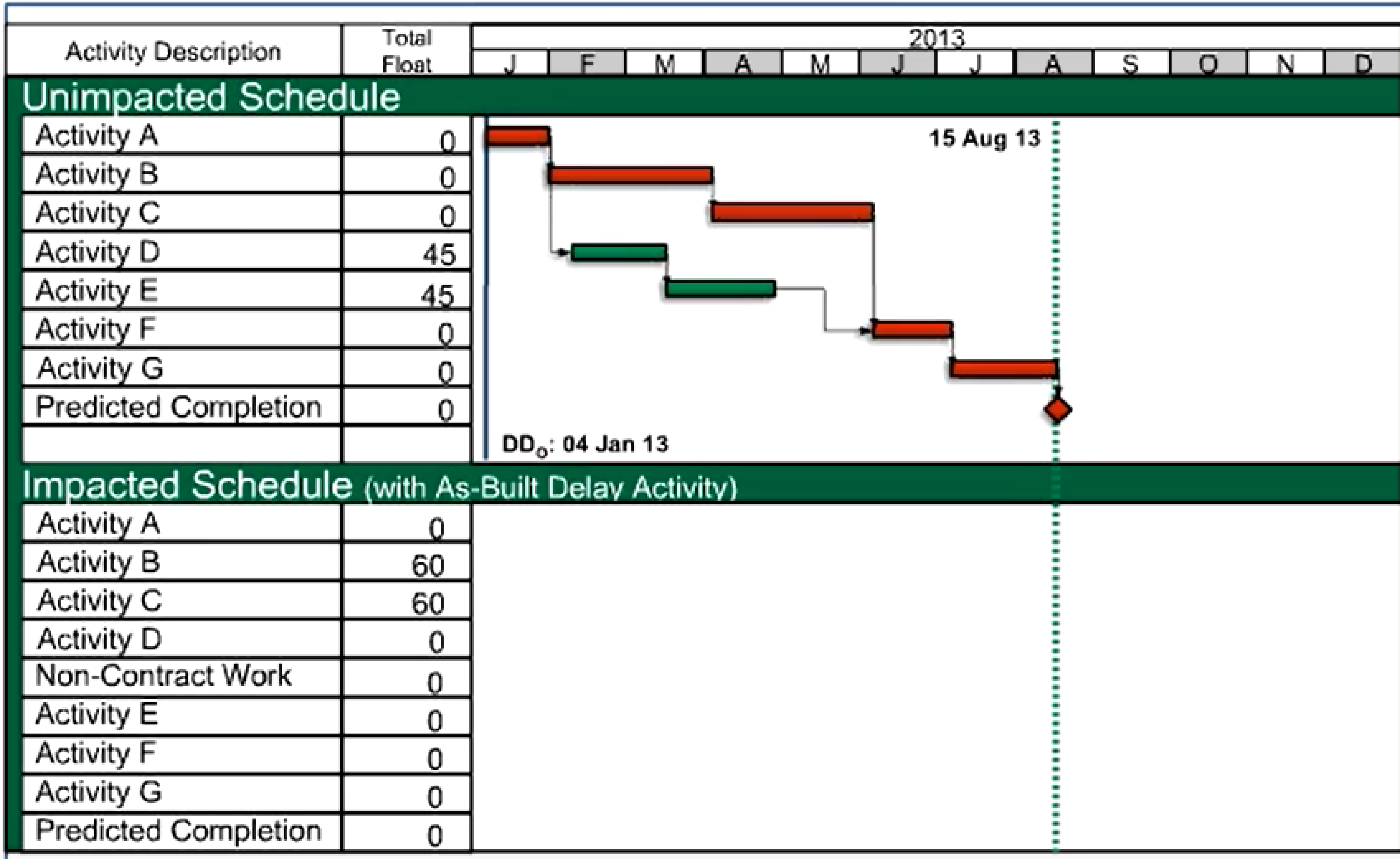
Unimpacted Schedule

- Impacted As-Planned uses the original baseline schedule as the unimpacted schedule.
- TIA typically uses one of the updated schedules, though implementations vary..

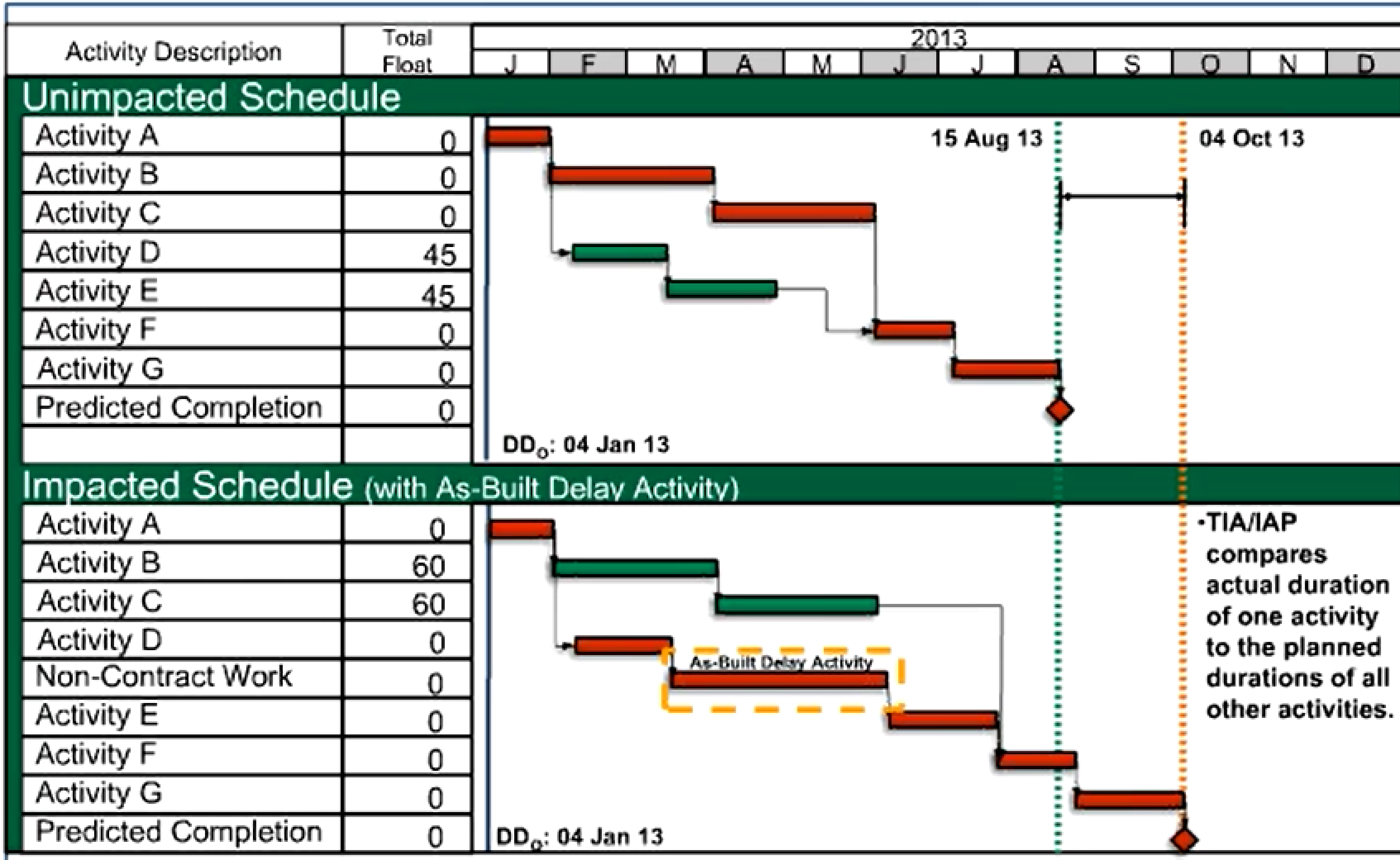
Impacted Schedule

- Schedule with an identical data date, and which is logically identical to the Unimpacted Schedule, except for the As-Built Activity.
- As-Built Activity is inserted into the network, logically tied into the impacted activity.

Retrospective time impact analysis – Example



Retrospective time impact analysis – Example



Retrospective time impact analysis – Example

General Information

Project Planned Start Date : 12,March,2012.

Project Planned Finish Date : 22,April,2013.

Available Updates Programme : 4 Programme

Retrospective time impact analysis – Example

- This has **<u>not been performed event by event</u>** because we don't know whether EDE or CDE occurred first in many places! **NON-INTEGRATED**

How to handle **concurrent delays**?

Window Technique – Case study- Window 0

Window	Window Start (1)	Window Finish (2)	Window Duration (3) = 2-1	Prog.	Source Programme	Project Finish date	Delay: Contractor	Delay: Employer	Total Entitlement
W0	10-Mar,2012	10-Apr,2012	31	P0	Appr. Baseline A.01	22,April, 2013	0	0	
				P1	P0+Delay Events	13,May, 2013		21	
				P2	P1+Progress update on DD	14,May, 2013	1		
Total Window Entitlement									22

Retrospective time impact analysis – Example

Window Technique – Case study- Window 1

Window	Window Start (1)	Window Finish (2)	Window Duration (3) = 2-1	Prog.	Source Programme	Project Finish date	Delay		Total Entitlement
							Contractor	Employer	
W1	11-Apr,2012	31-Dec,2012	264	P3	Approved P2	14,May, 2013	0	0	
				P4	P3+Delay Events	25,May, 2013		11	
				P5	P4+Progress update on DD	3,Jun , 2013	9	+	
Total Window Entitlement									20

Retrospective time impact analysis – Example

Window Technique – Case study- Window 2

Window	Window Start (1)	Window Finish (2)	Window Duration (3) = 2-1	Prog.	Source Programme	Project Finish date	Delay		Total Entitlement
							Contractor	Employer	
W2	1,Jan,2013	1,Feb,2013	264	P6	Approved P5	3,Jun , 2013	0	0	
				P7	P6+Delay Events	9,July, 2013		36	
				P8	P7+Progress update on DD	21,July, 2013	12		
Total Window Entitlement									48

Retrospective time impact analysis – Example

Window Technique – Case study- Window 3

Window	Window Start (1)	Window Finish (2)	Window Duration (3) = 2-1	Prog.	Source Programme	Project Finish date	Delay		Total Entitlement
							Contractor	Employer	
W3	2,Feb,2013	22,Apr,2013	79	P9	Approved P8	21,July, 2013	0	0	
				P10	P9+Delay Events	15,Aug, 2013		25	
				P11	P10+Progress update on DD	15,Sep., 2013	31		
Total Window Entitlement									56

Retrospective time impact analysis – Example

Window Technique – Case study- Summary

Window	Window Start	Window Finish	window Duration	Programme	Programme	Project Finish Date	Delay: Contractor Delay	Delay: Employer Delay	Total Entitlement
Window 0	10-Mar-12	10-Apr-12	31	P0	Approved BaseLine	22-Apr-13	0	0	
				P1	P0+Delay Events	13-May-13		21	
				P2	P1+Progress update on DD	14-May-13	1		22
window 1	11-Apr-12	31-Dec-12	264	P3	Approved P2	14-May-13			
				P4	P3+Delay Events	25-May-13		11	
				P5	P4+Progress update on DD	3-Jun-13	9		20
Window 2	1-Jan-13	1-Feb-13	31	P6	Approved P5	3-Jun-13			
				P7	P6+Delay Events	9-Jul-13		36	
				P8	P7+Progress update on DD	21-Jul-13	12		48
Window 3	2-Feb-13	22-Apr-13	79	P9	Approved P8	21-Jul-13			
				P10	P9+Delay Events	15-Aug-13		25	
				P11	P10+Progress update on DD	15-Sep-13	31		56
Total Entitlement							**53**	**93**	**146**

Retrospective time impact analysis – Pros and cons

- ✓ **Specifically designed** to measure project delay caused by **added or changed** work that was **not included in the project schedule**.
- ✓ **Sisters** to prospective time impact analysis
- ✓ Recognized by **Federal boards** in USA
- ✓ Can be **acceptable** analysis solution for **complicated networks**

Retrospective time impact analysis – Pros and cons

- × **Can** be **extremely complicated**
- × **Subject** to excessive **expert decision making**
- × **Susceptible** to **unintended** or **intended manipulation**
- × **Extremely sensitive** to the order of **fragnet insertion**
- × **True concurrency** and **primacy of delays** may be **ignored**
- × **Nature or cause** of the delay.
  - × Some delays may be caused by the **addition of work**
  - × **Suspension delays** occur because work cannot occur.
    - × **Despite this difference**, the Retrospective TIA treats these causes of delay the same.

Retrospective time impact analysis – When to use?

- When there are
  1. Logic linked baseline schedule
  2. Delay events to model
  3. Contemporaneous schedules with process information

In summary, the Retrospective TIA, though widely used, is problematic on many levels. If used, it should be used with ***great caution****.*

After-the-Fact Schedule Analysis

(Time Slice Windows Analysis (TSWA))

FSA methods

Method of Analysis	Analysis Type	Critical Path Determined	Delay Impact Determined	Requires
Time Slice Windows Analysis	Effect & Cause	Contemporaneously	Retrospectively	• Logic linked baseline programme. • Update programmes or progress information with which to update the baseline programme.
As-Planned versus As-Built Windows Analysis	Effect & Cause	Contemporaneously	Retrospectively	• Baseline programme. • As-built data.
Retrospective Longest Path Analysis	Effect & Cause	Retrospectively	Retrospectively	• Baseline Programme. • As-built programme.
Collapsed As-Built Analysis	Cause & Effect	Retrospectively	Retrospectively	• Logic linked as-built programme. • A selection of delay events to be modelled.

FSA methods

The Information Required					
	Baseline programme	Logic linked baseline programme	As-built data	Delay events to model	Up to date programmes/ process information
Time Slice Windows					
As-Planned v As-Built Windows					
Longest Path					
Collapsed As-Built					

The table above shows the evidence required for each delay analysis method. The table has been simplified.

Time slice windows analysis – Definition

- **Also** referred to **"snapshot"**, **"windows"** and **"contemporaneous period"** analysis
- Commonly used **observational method**
- It requires a **reliable series** of **contemporaneously updated baseline (TIME SLICES)** schedules reflecting an **accurate status** of the works at various snapshots (**being the time slices**).
- **Most experts** say that this is the **most accurate method**.

Why?

Time slice windows analysis – Definition

- **SCL Protocol:**
  - The **series of time slice programmes** reveals
    1. **Contemporaneous or actual critical path** in **each time slice period** as the works progressed and
    2. **Critical delay status** at the **end of each time slice**,

       thus allowing the analyst to conclude the **extent of actual critical delay** incurred within **each window**.

       *Thereafter, the **analyst investigates** the project records to determine **what events might have caused** the identified critical delay in **each time slice period.***

Time slice windows analysis – Definition

- AACE **method implementation protocol (MIP) 3.3, 3.4 and 3.5** say:
  - **Most often** performed **after project completion (retrospective)**, but **can** be done while project is **ongoing** as well.
  - **Compares** **two schedules** with **successive data dates** in order to determine the **driving critical activities**.
  - **Relies heavily** on the **contemporaneous** understanding of **criticality**
    - ✓ Because **every time** the **critical path** may **change**

Time slice windows analysis – How to use?

Pre-impact Schedule

- Most recently accepted schedule with a data date prior to the event.
- Schedule may also be updated to the day prior to the event.
- Does not contain the fragnet.

Post-impact Schedule

- Schedule which is logically identical to the Pre-Impact Schedule, except for progress.
- Does not need the fragnet (though it may be inserted with actual dates for record keeping).
- Recalculated as of the next appropriate data date.

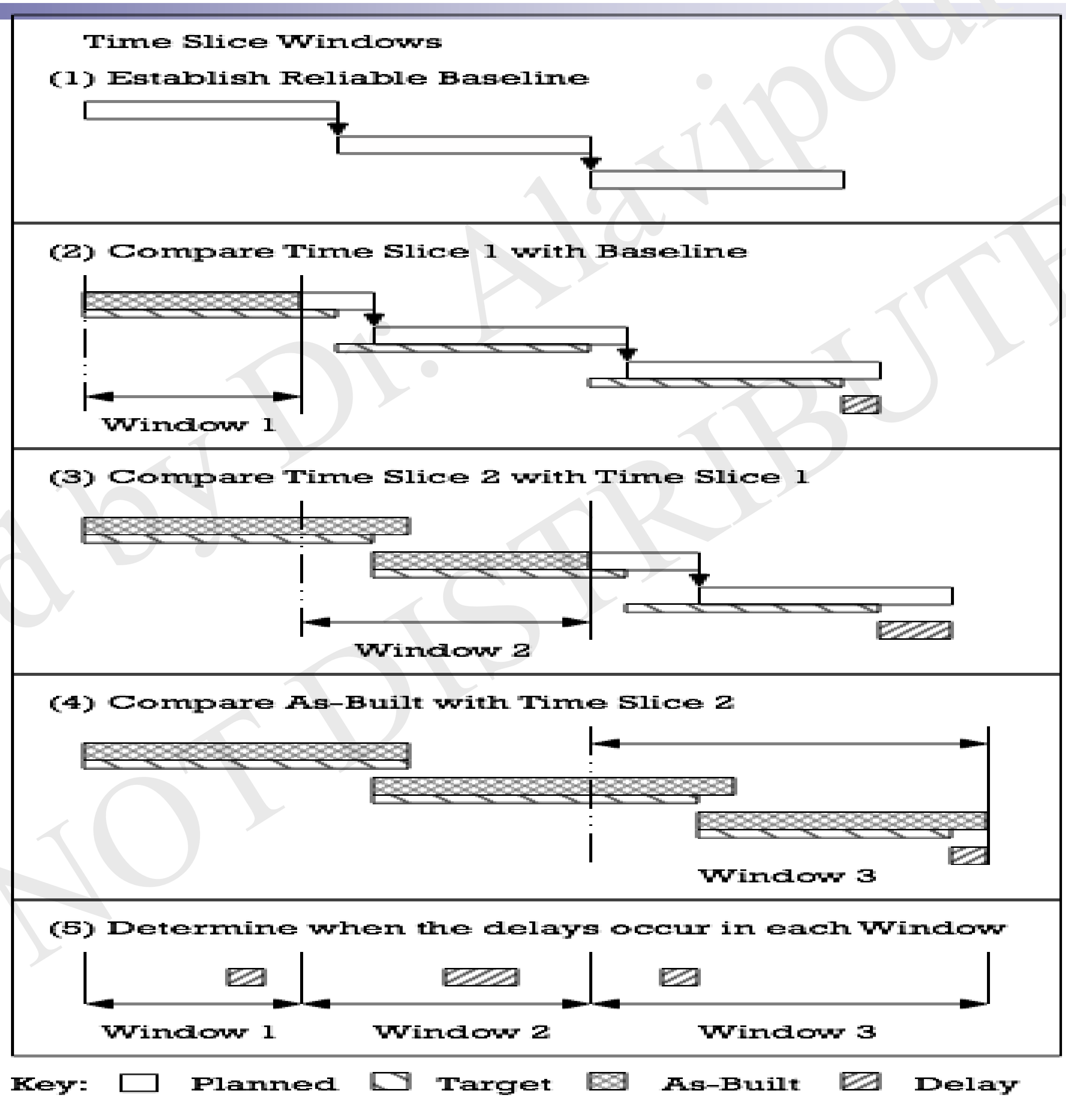
Time slice windows analysis – How to use?

1. **Logic linked baseline schedule** is **updated** at **regular intervals**, **usually monthly**.
2. **Each update** provides a **"snapshot"** status or **"time slice"** at the **point in time**.
3. The **analysis** at the **start** and **end** of **each window** are compared to assess
   1. **How much delay** occurred
   2. Identify **which activities** were **delayed**

   within each window.
4. Thereafter, the analyst **investigates** the **project records** to **determine** what **events** might have **caused** the **identified critical delay** in **each window**.

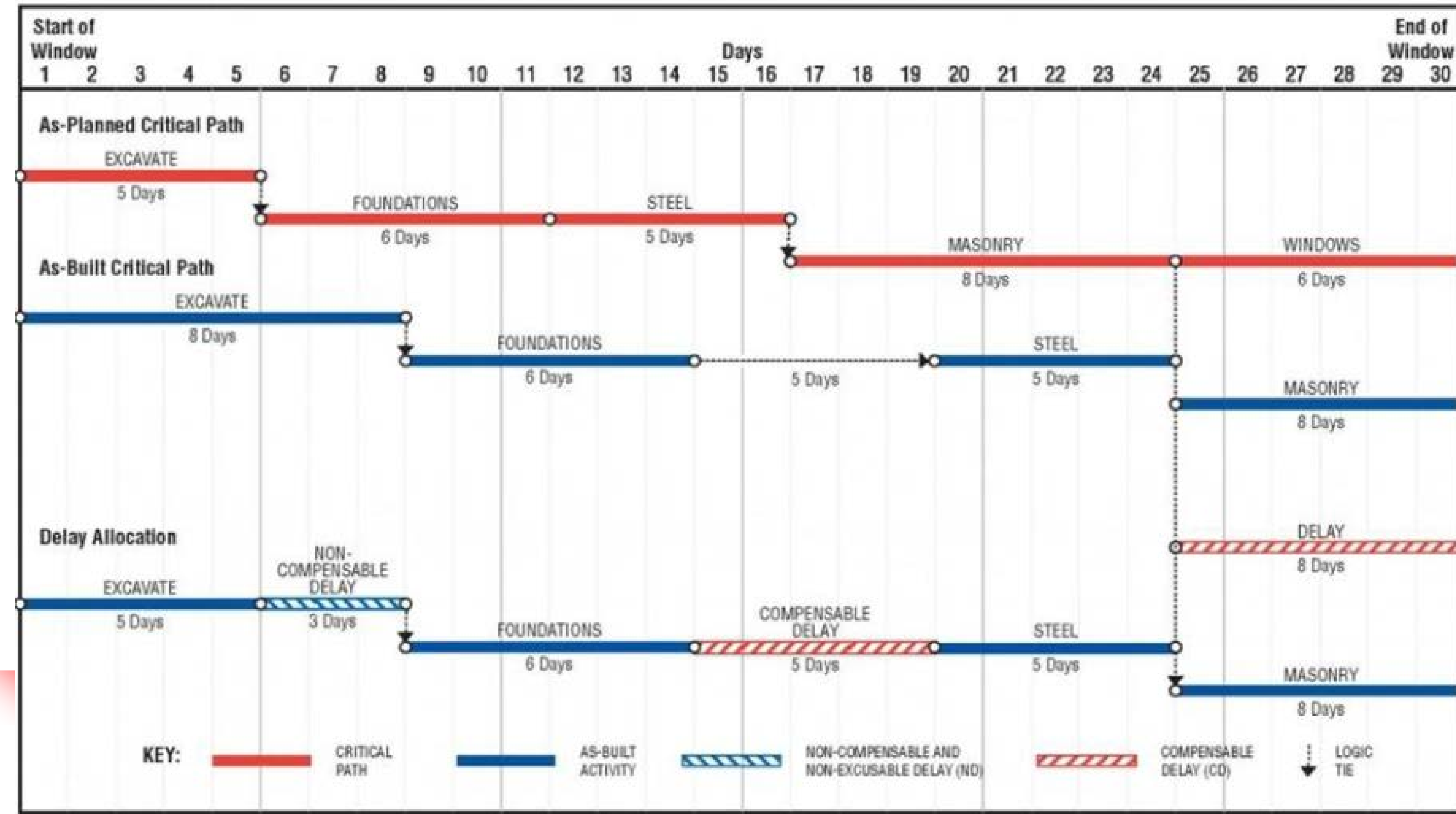
*For **each time slice schedule** the analyst **needs** to **verify** that the **historical components** reflect the **actual progress of the works**.*

Time slice windows analysis – Example

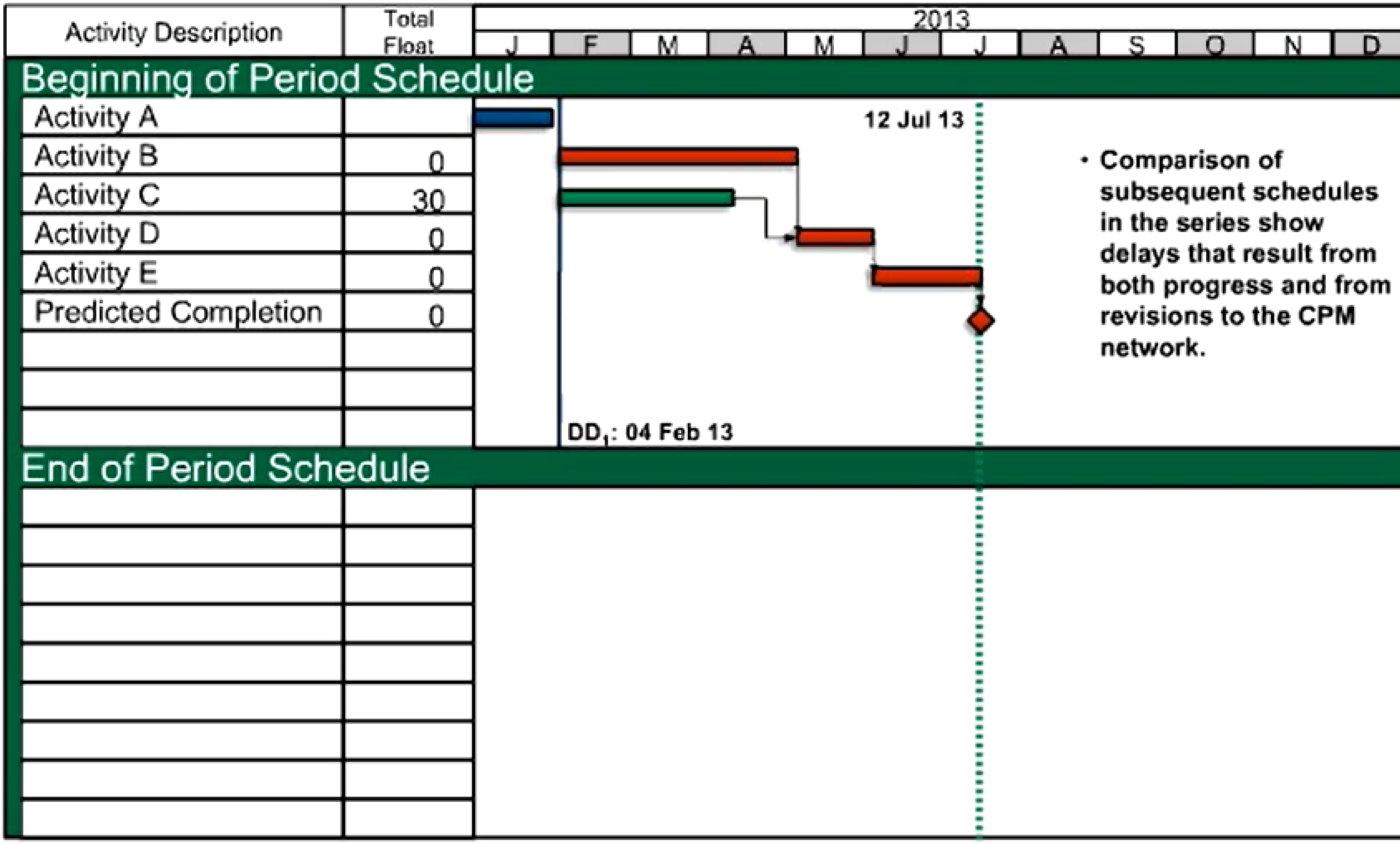


Time slice windows analysis – Example

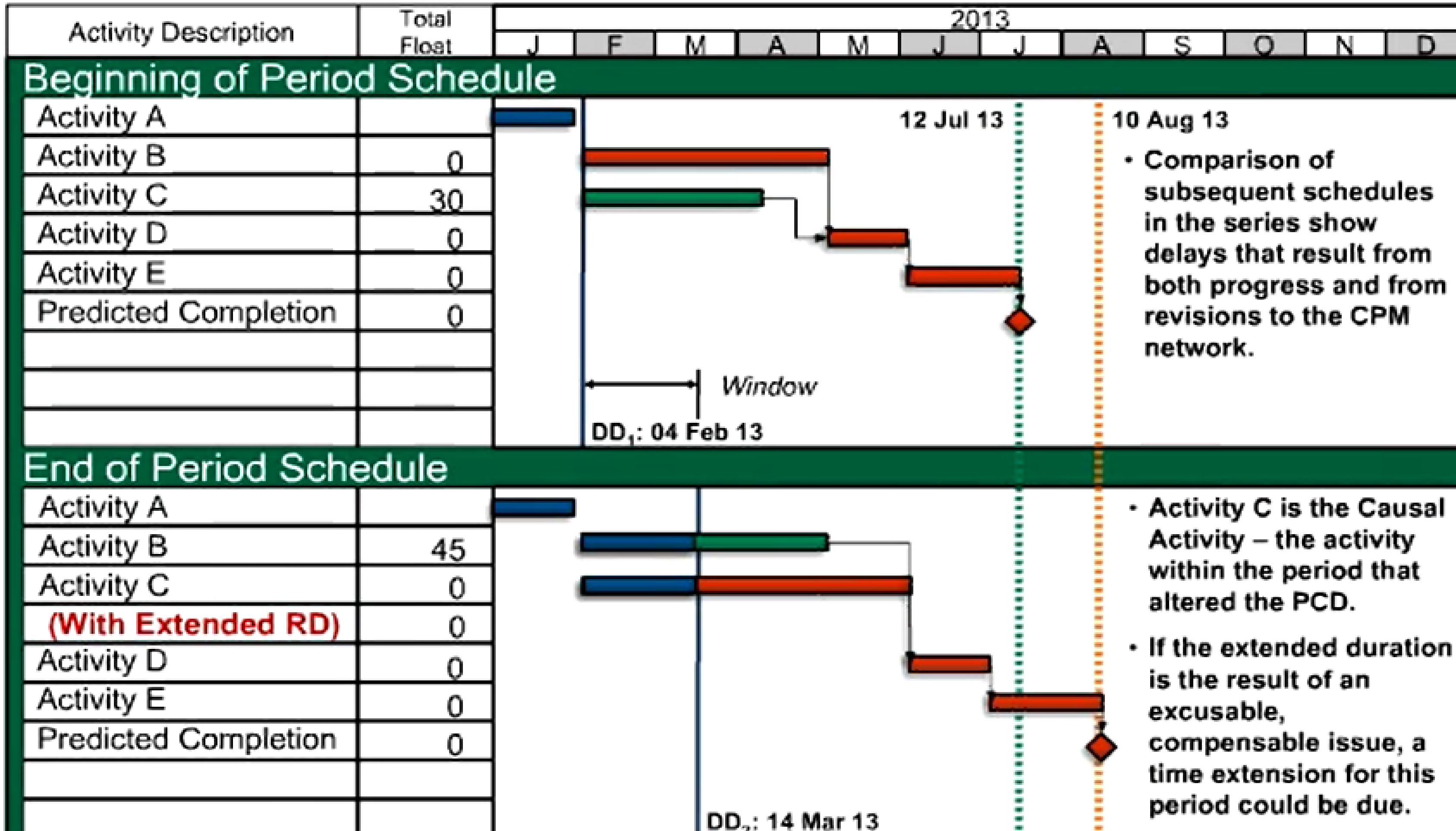
- Prepare the table of CDE, EDE, and concurrent from the as-built data.
  - For this purpose we need to adopt window-based approach on the occurrence of a delay



Time slice windows analysis – Example

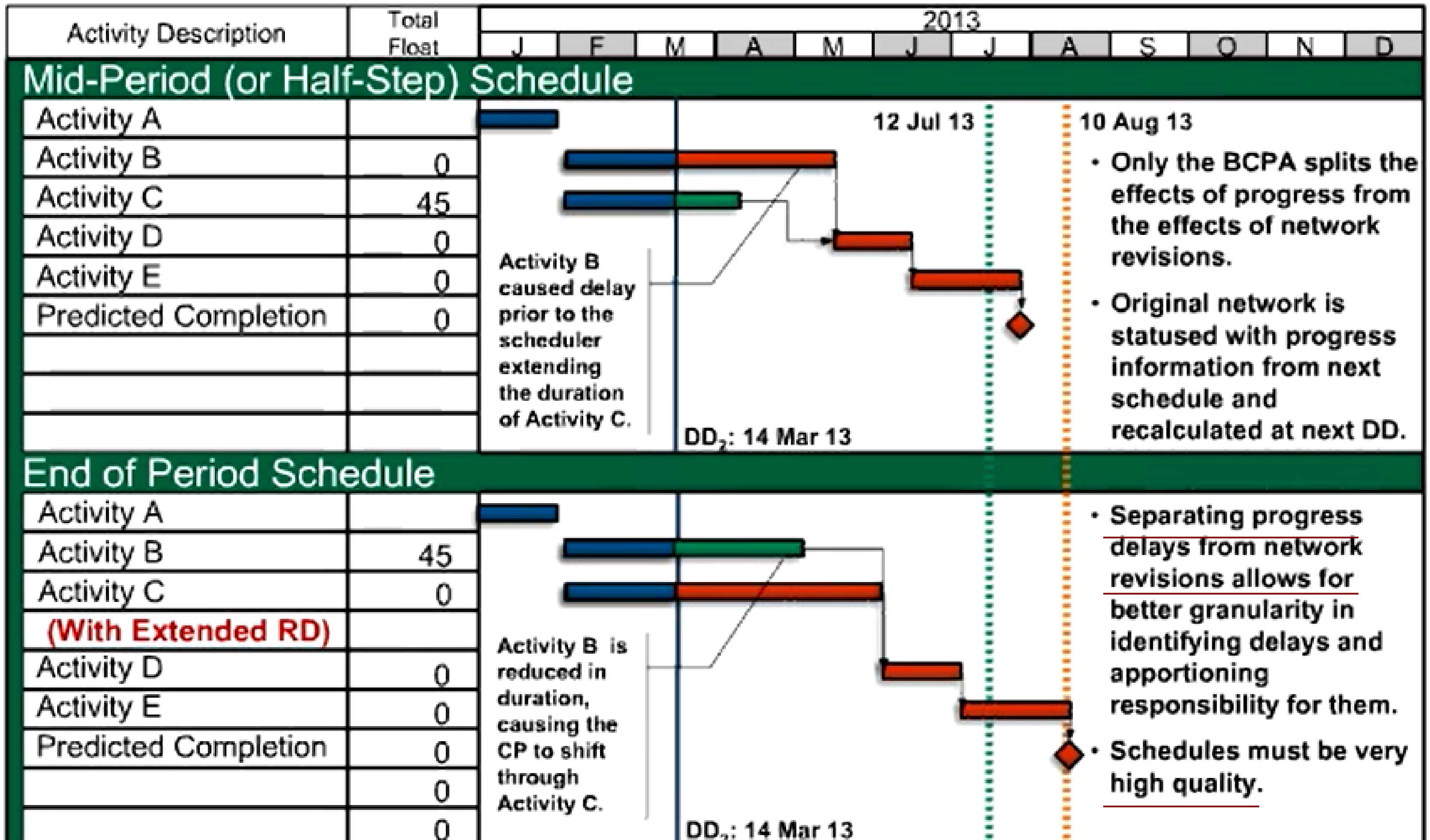


Time slice windows analysis – Example



Time slice windows analysis – Example

BCPA = BIFRUCATED CONTEMPORANEOUS PERIOD ANALYSIS



Time slice windows analysis – Example of court decision

Contractors need to anticipate that a court or board may express a preference for contemporaneously updated schedules over after-the-fact as-built schedules when assessing delay claims. The decision of the Armed Services Board of Contract Appeals, *Hedgecock Elec., Inc.*, ASBCA No. 56307, 12-2 BCA ¶ 35,086, illustrates this point.

Time slice windows analysis – Pros and cons

- ✓ Divides the project into manageable time periods, and therefore reflects the impacts on periodic basis.
- ✓ **Extremely accurate** if: (1) **accurate updates (contemporaneous schedule)** and (2) **bifurcated methodology**
  - ✓ **But not always** all of them
- ✓ **Probably** the **most** widely **used** by **experts**
- ✓ **Commonly accepted retrospective** analysis technique by almost **all courts**
- ✓ **Dynamic analysis** (**contemporaneous critical path**)
- ✓ **Can** be used for **after-the-fact** schedule (**retrospective delay impact**)
- ✓ It **considers actual progress** and **changes (not hypothetical)**
- ✓ **Concurrent delays** can be **considered**
  - × If windows are large, it could also cause a problem

Time slice windows analysis – Pros and cons

- **Contemporaneous** schedule updates must be **validated** as **accurate**
  - **Systematic schedule updates** are required.
- **High** in terms of **cost** and **time** (more than other methods)
- **Subjective** in that choice of **window affects outcome**
    - If **one window** (**update**) is **not accurate**, the **result** will be **subjective**
- It **requires** a **detailed data** to analyze

Time slice windows analysis – When to use?

- When we have a **reliable series** of **contemporaneously updated baseline** schedules reflecting an **accurate status** of the works at **various snapshots** (being the **time slices**). It has **three criteria**:

1. The **logic linked baseline** must be **detailed** and **no flaws** be in the schedule

2. The **historic element** of each **snapshot schedule** must be developed from **accurate** and **detailed as-built data**
   - If **regular detailed progress** data is **not available**, then it is **highly unlikely** that this method can be **used**

3. The **future element** of **each snapshot schedule** should **accurately** and reasonably represent the **status** of the **works** on the **analysis date**.

Time slice windows analysis – Practical procedure

- The **steps** for performing a **contemporaneous period analysis** are summarized below:

1. Become familiar with the project and relevant documents
2. Perform a high level review of the available project schedules
3. Select schedule windows
4. Identify the critical and near-critical paths
5. Perform a detailed review of the schedules selected for the analysis
6. Determine the changes made between the schedules selected for the schedule windows
7. Make copies of the analysis schedules and implement necessary corrections
8. Develop variance tables to calculate date and duration variances
9. Research activity impacts and allocate responsibility for delays
10. Sum and document the results

Time slice windows analysis – Practical procedure

1. Become familiar with the project and relevant documents

- **Review** the **contract** and **understand** the **contract documents**
- **Note** any **contract provisions** pertaining to a **claim situation**
- **Understand** the **contract completion dates**
  - **Specify** if the **contract** requires **any interim completion milestones** **prior** to **overall project completion**.
- **Evaluate** **change orders** to **understand** how the **contract** has been **modified** and **if any time extensions** have been **executed** as part of any **change orders**.
- **Review** **contract scope of work** and **know** the **parties** to the **contract**.

Time slice windows analysis – Practical procedure

2. **Perform a high level review of the available project schedules**

- **Electronic schedules** should be provided.
- **Analyst** should **review** the following:
    1. **Available schedules**
    2. **Number of activities** and **relationships** in each available schedule file
    3. **Changes** that **occur between** each of the **contractor's schedule updates**

Time slice windows analysis – Practical procedure

2. Perform a high level review of the available project schedules

- **Contractors** are required to provide **monthly schedule updates**, **but additional updates** may be **available** that model **certain impacts** or "**what-if" scenarios** completed by the contractor.
  - Additionally, a **prudent contractor** on a **large** and or **difficult project** may elect to:
    1. **Update** the schedules **more than once a month**
    2. **Update** a schedule in **conjunction** with a **large impact or change order**

Time slice windows analysis – Practical procedure

2. Perform a high level review of the available project schedules

- **After initial review**, **several schedule** should be selected for a **more thorough review**.

<u>Items to review and note include</u>:

1. Activity ID structure
2. Level of detail in activity names
3. Existence and quality of a work breakdown structure and/or activity coding
4. Open-ended activities
5. Number and types of activity constraints
6. Calendars used and the types of activities assigned to each calendar
7. Cost and resource loading assignments

<u>Determine</u> which activity will be used to track the <u>completion of the project</u>.
It can be final or substantial completion.
<u>This activity</u> will be used in the delay analysis to quantify delay in each schedule window.

Time slice windows analysis – Practical procedure

3. Select schedule windows

- **Now**, the **analyst** has an **understanding** of **contract requirements** and **project schedules**.
- The **analyst now** should **select** the **schedules** that will be used as the **beginning** and **end** of the **schedule windows**.
    - **These windows** become the **periods of time** that will be **studied** in the analysis.

Time slice windows analysis – Practical procedure

3. Select schedule windows

Timing of significant milestones, impacts, and delay events are important factors to select the beginning and end of the schedule windows.

The entire project may not need to be part of the analysis

- **AACE says:**

Every update may not be used. However, every update should be considered. Accuracy is reduced if updates are not evaluated for multiple-month periods when schedule revisions or changes in sequence occurred to the project plan.

Time slice windows analysis – Practical procedure

3. Select schedule windows

- Identify the **contractor's as-planned CPM programme** and **contemporaneous progress updates** through to completion.
- **Once the schedules are selected**, the schedule **data dates** establish the **beginning** and/or **ending dates** of the **schedule windows**.

Table 1
Sample Schedule Windows

Window No.	Schedule Window Dates	Schedule Data Date (Beginning of Window)	Schedule Data Date (End of Window)	Length of Window (Cal. Days)
1	21 October 2013 – 28 January 2014	21-Oct-2013	29-Jan-2014	101
2	29 January 2014 – 20 March 2014	29-Jan-2014	21-Mar-2014	52
3	21 March 2014 – 27 July 2014	21-Mar-2014	28-Jul-2014	130
4	28 July 2014 – 25 October 2014	28-Jul-2014	26-Oct-2014	91
5	26 October 2014 – 5 December 2014	26-Oct-2014	06-Dec-2014	42

Time slice windows analysis – Practical procedure

4. Identify the critical and near critical paths

a) Based on total float:

- ❖ Determine the **most critical activities** in **each window** by identifying the activities with the **least amount of positive total float** ***(near critical path)***
- ❖ Determine the **most critical activities** in **each window** by identifying the activities with the **greatest amount of negative total float**

b) Based on longest path:

- ❖ Tasks which were **predecessors to “driving” critical activities** along each path

c) Concurrent as-built critical paths

Time slice windows analysis – Practical procedure

4. Identify the critical and near critical paths

- **Review** and **document** the **critical** and **near-critical paths** for selected **schedule windows**.
- The **paths** then must be **checked** for **reasonableness** (**THIS WILL BE TALKED IN NEXT STEP**).
  - **Constraints**, **lags** and **logics** should be **checked** **before** the **analysis**
  - **Analyst** should **document** these **issues** and **identify** if **corrections** to any activities are necessary.

The coding of the critical and near-critical paths is an important step!

This step is often omitted due to time and budget constraints because it is time consuming to document and determine the actual dates and as-built schedules.

Time slice windows analysis – Practical procedure

5. Perform a detailed review of the schedules selected for the analysis

- **Upon <u>reviewing</u>** and **<u>coding</u>** the **critical** and **near-critical paths**,

  each schedule should be **reviewed** in **<u>detail</u>**

  to determine **if any schedule corrections** are necessary.

- It is **important** to:

  1. **<u>Evaluate</u>** the **quality** and **reasonableness** of the **schedules**
  2. **<u>Identify</u> items** that may be **necessary** to **correct** to ensure **schedule accuracy**

Time slice windows analysis – Practical procedure

5. Perform a detailed review of the schedules selected for the analysis

- **Typical items** that can **affect** the analysis include**:**
  1. Excessive number of open-ended activities
  2. Overuse of constraints
  3. Activity and logic changes
  4. Duration changes
  5. Calendar changes
  6. Activities with unreasonably large total float values
  7. Inconsistent use of schedule calculation methods
  8. Incorrect as-built dates
  9. Excessive positive or negative relationship lags

Any deficiencies in any above items should be corrected before critical and near-critical paths are found.

Let's talk about some of them!

Time slice windows analysis – Practical procedure

5. **Perform a detailed review of the schedules selected for the analysis**

- **Open-ended activities**
  - One of the **most fundamental quality checks**
  - **Activities** that **lack** a **predecessor**, **successor**, or **both**
  - **Exception** to this is the **project start** and **completion activities**
  - A large number of **open-ended activities** will **create** **float values** and **forecasted completion dates** that may be **incorrect** and they **MUST BE CORRECTED**.

Time slice windows analysis – Practical procedure

5. Perform a detailed review of the schedules selected for the analysis

- **Overuse of constraints**
  - Use of a constraint **restricts** the **start or finish of an activity**.
  - The **use of a few soft constraints** is **not bad practice**, however **a large number** of constraints should be **avoided**.
    - This **restricts** the **normal forward and backward path** scheduling calculations.
    - The **misuse** of constraints may **generate** **artificial activity dates** and **float values**.

The analyst must evaluate the relevance of the constraint to determine

(1) Reason for its use

(2) If the constraint should be removed from the schedule

Time slice windows analysis – Practical procedure

5. Perform a detailed review of the schedules selected for the analysis

- **Use of calendars**

  - **Several aspects** of the project calendars **must be considered**.

1. **Calendars themselves should be considered.**
   - **Note** if the **specific calendars** were **changed** **during the evolution** of the schedules.
     - A five-day, eight-hour per day calendar may have been changed to a five-day, ten-hour per day calendar.

2. **Note if the calendars include planning for adverse weather.**
   - A **properly developed schedule** will include **anticipated lost time** due to **inclement weather**.
     - This should be based on **contractual requirements** and **historic weather data** specific to the geographical area of the project

3. **Determine if activities had a calendar change during the selected schedule window**

Changes between schedule updates are discussed in next step

Time slice windows analysis – Practical procedure

5. Perform a detailed review of the schedules selected for the analysis

- **Other common schedule deficiencies include the following:**
  - Incorrect and inconsistent use of as-built dates;
  - Activity planned durations that are inconsistent with the Contractor's bid estimate calculations;
  - Missing contractual scope of work;
  - Lack of contractually required completion activities;
  - Overuse of constraint dates;
  - Missing predecessor or successor logic ties creating open-end activities;
  - Inadequate depiction of equipment and materials procurement and delivery activities;
  - Inaccurate predecessor logic for Owner approval activities;
  - Lack of a contractual Owner approval period for submittal and turnover packages;
  - Lack of reasonable project punch list and demobilization periods;
  - Inconsistent use of Calendars, Progress Override and Retained Logic schedule calculation options; and
  - Unrealistic as-built logic relationships indicating out-of-sequence progress.

Time slice windows analysis – Practical procedure

6. Determine changes made between the schedules selected for the schedule windows

- In **addition** to **deficiencies**, **changes** made between **schedule updates** should be identified.
  - Any **delays** or **acceleration** efforts that **added** or **removed** should be identified.

- **Three common changes:**
  - Logic changes
  - Activity changes
  - Duration changes

Time slice windows analysis – Practical procedure

6. Determine changes made between the schedules selected for the schedule windows

- **Logic changes**
  - Analyst **must** document:
    1. Any **activity additions** to schedule **logic**
    2. Any **activity subtractions** from schedule **logic**
    3. Any **modifications** to schedule **logic**

Time slice windows analysis – Practical procedure

6. Determine changes made between the schedules selected for the schedule windows

- **Activity changes**
  - **Check** the **consistency** of **activity IDs** and **names**

If activities are continually changing IDs and names, the analyst may conclude that the contractor's scheduling was not in accordance with industry accepted and standard practices.

In this case, it is very difficult to perform the analysis

Time slice windows analysis – Practical procedure

6. Determine changes made between the schedules selected for the schedule windows

- **Duration changes**
  - **Contractors** may **modify** the **scope** of an activity by adjusting the **original or remaining duration**

Although this is not the good practice, if the analyst notes a delay to an activity that was caused by increased original or remaining duration of an activity, the changed duration will need to be researched to determine if the increase was justified.

When faced with delays in the schedules, many contractors will simply reduce activity durations to mitigate delay.

This is often done without any basis. These instances must be noted for the analysis.

Time slice windows analysis – Practical procedure

7. Make copies of the analysis schedules and implement necessary corrections

- **Each of the corrections** identified in **prior analysis steps** is **implemented** in **this step** and **each change** must be made in **all of the analysis schedules**.

- **Analyst** should make **copies** of the **original schedule files**, **prior** to making **changes**.
  - **Any changes** to the schedule should be **made** in **copies of the original files.**

- The corrections to the schedules are an acceptable part of AACE 29R-03:

  *The **issue of correcting** the schedule is one of **balance** and **reasonableness** and, for this reason **corrections** should **not be made across the board** or **automatically!***

Time slice windows analysis – Practical procedure

7. Make copies of the analysis schedules and implement necessary corrections

- **Review critical and near-critical paths**
  - The **critical** and **near-critical paths** must be **re-evaluated** to **determine whether if** the **corrections** or **modifications** to schedule cause any **changes** to **critical or near-critical paths**.
    1. If no changes, analyst can proceed to next step
    2. If changed, additional research is required
       - The change should be identified.
       - The validity of the change must be reevaluated.
       - The nature of the change should be discussed and identified.

*In some cases, if the **owner** is **uncertain** and the **project team** is **unavailable** (as is sometimes the case in forensic analysis), the **analyst** must **make a decision** on the **best option** and which **critical path** reflects the work as of the **data date**.*

This decision must not be made in the best interests of the owner, but must reflect the most accurate portrayal of the project record.

Time slice windows analysis – Practical procedure

8. Develop variance tables to calculate date and duration variances

- **After identification** and **confirmation** of the **critical and near-critical paths**, analyst should **develop variance tables** for **each window** which calculate
  1. **Date variances** for **each activity** within the **schedule windows**
  2. **Duration variances** for **each activity** within the **schedule windows**

Why we need the variance calculations?

*Because **delays** can be the **result** of*

1. *An activity **not starting on time** and/or*
2. *An activity **taking longer than planned**,*

*it is **necessary** to perform **variance calculations** for the **date** and **duration** of the activities.*

Time slice windows analysis – Practical procedure

8. Develop variance tables to calculate date and duration variances

- For this step, **analyst** typically **exports** the **activities** and **associated data** to a **spreadsheet.**

Keep a record of the float values for all of the driving activities in a "float map" which tracks the gain and loss of float against each driving activity through all of the the available contemporaneous progress records.

Time slice windows analysis – Practical procedure

8. Develop variance tables to calculate date and duration variances

- **Variance tables** should be developed using the **spreadsheets** for the **two schedules** which correspond to the **beginning** and **end** of a **schedule window**.
  1. **Each spreadsheet** will be the **summary** of the **window**
  2. **Each spreadsheet documents** the **variance** (**gain** or **delay**) for each activity's
     - Original duration
     - Remaining duration
     - Start date
     - Finish date

Time slice windows analysis – Practical procedure

8. Develop variance tables to calculate date and duration variances

Table 2
Sample Variance Table

Activity ID	TF (Days)	Activity Status	LI CP	05-Sep-13 OD (Days)	05-Oct-13 OD (Days)	OD Var. (Days)	05-Sep-13 Start Date	A	05-Oct-13 Start Date	A	Start Var.
D1590	-51	Not Started	CP1	8	8	0	07-Oct-2013		14-Nov-2013		38
M1500	-45	Not Started		1	1	0	13-Sep-2013		22-Oct-2013		39
M2010	-43	Not Started		6	4	-2	11-Sep-2013		11-Oct-2013		30
S1480	-51	In Progress	CP1	10	13	3	24-Jun-2013	A	24-Jun-2013	A	0
S1620	-45	Not Started		5	12	7	07-Sep-2013		17-Oct-2013		40
S1760	-38	In Progress		9	9	0	03-Sep-2013		05-Sep-2013	A	2
S1835	-51	In Progress	CP1	6	6	0	28-Jun-2013	A	28-Jun-2013	A	0

TF – Total Float   CP – Critical Path   OD – Original Duration   A – Actual   Var. – Variance

- Finish variance and remaining duration variance not shown.
- Activity name has been omitted , but should be included.
- This should be prepared for each window.

Time slice windows analysis – Practical procedure

8. Develop variance tables to calculate date and duration variances

- **After** the **table** is **developed**, the **spreadsheet** should be **filtered** for the **critical and near-critical activities** to determine which activities are the **causes of delays** in the **window**.

- The **activities** that are **found** to be the **causes of delay** in **each window** may be **referred** as **critical path originating activities**.

- **These activities** must be **researched** to **determine** the **allocation of delay**

Time slice windows analysis – Practical procedure

9. Research activity impacts and allocated responsibility for delays

Investigate the cause of delay to the driving activities in each period where a measurable loss or gain is identified

- For **each critical path identified** in the **prior step**, the **schedules** and **contemporaneous project documentation** must be **researched** to:

  determine the **factor or factors** that may have **caused** the **start delay** and/or **extended activity duration**.

- **Activities** and **logic** may have been **modified** as **acceleration measures**, or **delay** may have been added via **duration adjustments**.
  - **If** these instances are **noted**, the **project record** must be **analyzed** to determine **if** these **changes** were **supported**.

Time slice windows analysis – Practical procedure

9. Research activity impacts and allocated responsibility for delays

- **Determine** the **factor or factors** that **may have caused** the start delay and/or extended activity duration
- A **sample** of **documented critical paths**, **originating activities**, and **delay summary** are included in the table below:

Table 3
Critical Paths and Originating Activities

Critical Path	Activity ID	Activity Description	Delay Summary
1st Critical Path	A1800	Excavate for Sump A19	22 day finish delay
	A1790	Install Sump A19	15 day start delay due to excavation – OD/RD decreased from 12 to 4 days
	A1810	Excavate for Underground Sump Tank	5 day OD/RD decrease
2nd Critical Path	P1825	Complete Pipe delivery for 30" Raw Water Line	36 day finish delay
	P1775	Install Raw Water Piping	15 day finish delay
	P1750	Backfill Raw Water Piping	8 day start delay and 25 day finish delay

OD – Original Duration   RD – Remaining Duration

How to allocate responsibility? Let's see my approach!

Time slice windows analysis – Practical procedure

9. Research activity impacts and allocated responsibility for delays

1. **If there is <u>one type of delay</u> in each window (EDE, CDE, or concurrent)**

   a) <u>If Delays to the originating activities > total delay recognized in the window</u>

      the **contractor most likely** implemented **<u>acceleration measures</u>** in the schedules for the **window**.

      ❖ Although the **<u>maximum delay is 36 days</u>**,

        if the **total delay** was **ten days during the window**,

        the analyst would allocate the <u>ten days of delay, not 36 days</u>.

   b) <u>If Delays to the originating activities = total delay recognized in the window</u>

      You can easily allocate all delays

   c) <u>If Delays to the originating activities < total delay recognized in the window</u>

      There should be a problem and you should review the result

What if there are <u>two or more types of delays</u> in a window?

Time slice windows analysis – Practical procedure

9. Research activity impacts and allocate responsibility for delays

2. **If there two or more types of delays in each window (EDE, CDE, or concurrent)**
   a) If you have updates immediately **before and after** the time when each different type of delay occurs ***(bifurcated updates)***

There would be no difficulty to calculate (similar to Case 1)!

Time slice windows analysis – Practical procedure

9. Research activity impacts and allocate responsibility for delays

2. If there two or more types of delays in each window (EDE, CDE, or concurrent)

b) If you do not have bifurcated updates and other delays also exist in a window

i. **If EDE (10) + CDE (5) = total delay recognized in the window (15)**

No concurrent delays = Allocate based on each delay

ii. **If EDE (10) + CDE (5) > total delay recognized in the window (12)**

a) Primacy of delays = One that occurred first may have created float for others

- *You may perform **collapsed as-built analysis***

*b) Concurrent delays = May concurrent delay exists in the window **(3).***

- *If you don't know the **time of occurrence**, you need to accept functional theory and overlook primacy of delays*

EDE = 10 -3 = 7 & CDE = 5 – 3 = 2, Concurrent = 3

iii. **If EDE (10) + CDE (5) < total delay recognized in the window (17)**

There should be a problem and you should review the result

Time slice windows analysis – Practical procedure

9. Research activity impacts and allocate responsibility for delays

Apportion responsibility for the loss or gain measured in each window by reference to the driving activity and causative events documented in the contemporaneous records.

Time slice windows analysis – Practical procedure

9. Research activity impacts and allocate responsibility for delays

- It is also **necessary** to research any **potential weather impacts** during the **window** to **determine** if **part**, or **all**, of the delay is **due to adverse weather**.

- **Lost time** due to **inclement weather** may change the **apportionment of delay** **if** the **weather delay** is **concurrent** with a **contractor delay**.

- **Weather delays** that are **excusable**, may **not be compensable**, **however** the **contract terms** should be **reviewed** **prior** to **allocating delay** due to weather.

Time slice windows analysis – Practical procedure

10. Sum and document the results

- The **analyst** must perform these steps and allocation for **each of the schedule windows**.

- **Upon completion** of the exercise, the **total delay allocation** should be **summed** and **documented**.

After-the-Fact Schedule Analysis

(As-Planned vs. As-Built Analysis (APAB))

FSA methods

Method of Analysis	Analysis Type	Critical Path Determined	Delay Impact Determined	Requires
Time Slice Windows Analysis	Effect & Cause	Contemporaneously	Retrospectively	• Logic linked baseline programme. • Update programmes or progress information with which to update the baseline programme.
As-Planned versus As-Built Windows Analysis	Effect & Cause	Contemporaneously	Retrospectively	• Baseline programme. • As-built data.
Retrospective Longest Path Analysis	Effect & Cause	Retrospectively	Retrospectively	• Baseline Programme. • As-built programme.
Collapsed As-Built Analysis	Cause & Effect	Retrospectively	Retrospectively	• Logic linked as-built programme. • A selection of delay events to be modelled.

FSA methods

The Information Required					
	Baseline programme	Logic linked baseline programme	As-built data	Delay events to model	Up to date programmes/ process information
Time Slice Windows					
As-Planned v As-Built Windows					
Longest Path					
Collapsed As-Built					
The table above shows the evidence required for each delay analysis method. The table has been simplified.					

As-planned vs. as-built windows analysis – Definition

- This is the **second** of the **“windows”** analysis methods.
- It is **less reliant** on **programming software**
- The **prerequisite** for this type of analysis is:
  1. **A baseline schedule**
  2. **An as-built schedule**
     - **As-built schedule** can **either** be:
       1. An **existing** as-built schedule or
       2. An as-built schedule **constructed** from **scratch**

In either case the **as-built schedule** needs to be **verified** using the **as-built data** however **no specific guide** is given in the **Protocols**.

As-planned vs. as-built windows analysis – Definition

- AACE **method implementation protocol (MIP) 3.1 and 3.2** say:
  - **Compares** the **as-planned schedule** to the **as-built**, establishes an **as-built critical path**, and **determines** what **events** actually **drove project completion**
    - ✓ If windows analysis performed (periodic) = contemporaneous critical path
    - ✓ If total time analysis performed (gross) = as-built critical path

As-planned vs. as-built windows analysis – How to use?

As-Planned Schedule

- Accepted/approved baseline schedule from the project.
- Unaltered CPM network.

As-Built Schedule

- Either the fully statused schedule from the project, or a schedule recreated from project records (Daily Specific As-Built).
- Contains both parties' delays.
- Often associated with "total time" delay claims, which are rejected by the courts.
- Sophisticated implementations are not total time.

As-planned vs. as-built windows analysis – How to use?

1. The **duration** of the work is **broken** down into **windows**.
   - ❖ **Time slices** are **not necessarily used**.
     - ✓ **Regular intervals not** needed.
   - ❖ Instead, **windows** are defined as **periods** between the dates when **key items** were **actually started** or **achieved**.
     - ✓ Sometimes we don't have update at the time of key items, sometimes have
2. **Windows** are **framed** by **revised contemporaneous schedules**, **contemporaneous update schedules**, **milestones**, **delays** or **significant events**.
   - ❖ Analyst determines the **contemporaneous** or **retrospective (as-built) critical path** in **each window**

     by a **common-sense** and **practical analysis** of the **available facts**.
     - o As this task does **not substantially rely** on **programming software**,

       it is important that the **analyst** sets out the **rationale and reasoning** by which **criticality has been determined**.

As-planned vs. as-built windows analysis – How to use?

3. Find the **contemporaneous** or **retrospective (as-built)** critical path
4. **Each window** is then **determined** by comparing **key dates** along
   - The **contemporaneous** or **retrospective (as-built)** critical path

     against

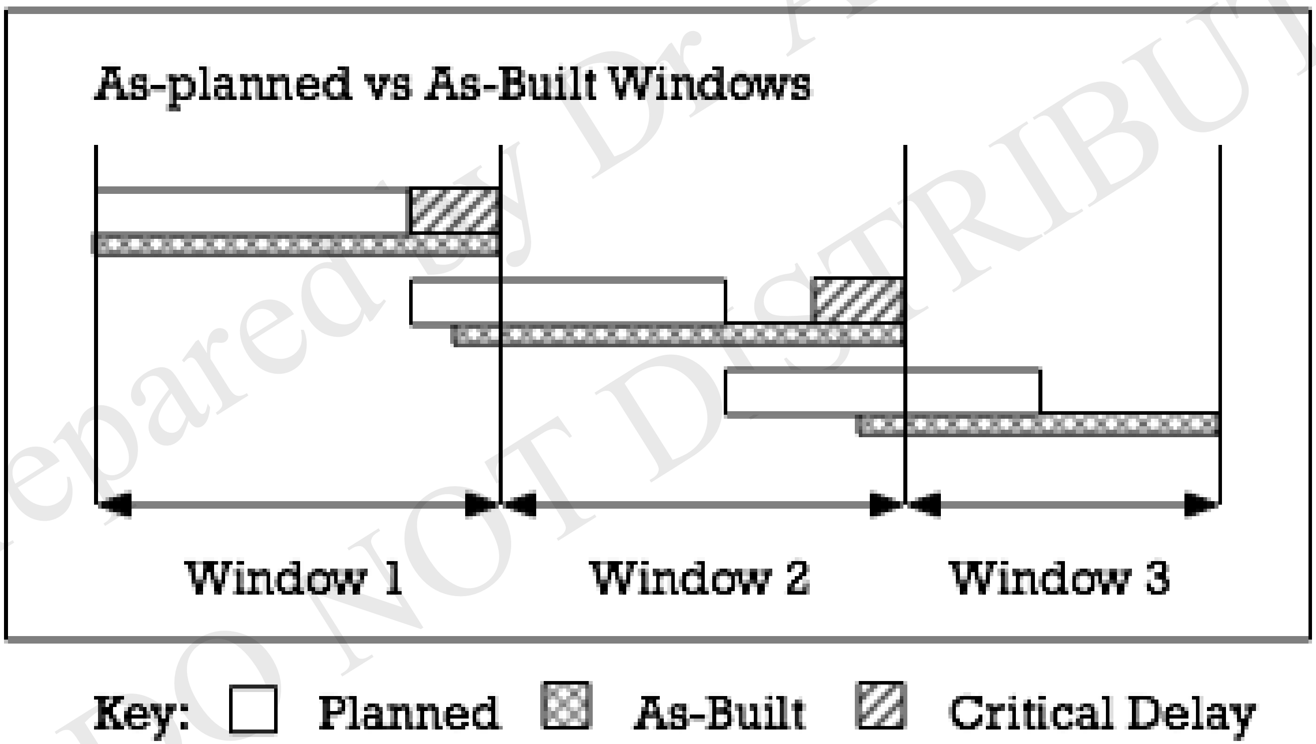
   - Corresponding **planned dates** in the **baseline schedule**.

5. Thereafter, the **analyst investigates** the **project records** to **determine** what delay **events** **might have caused** the **identified critical delay**

6. The **critical delay** incurred and the **mitigation** or **acceleration** achieved in **each window** is **accumulated** to identify **critical delay** over the **duration of the works**.

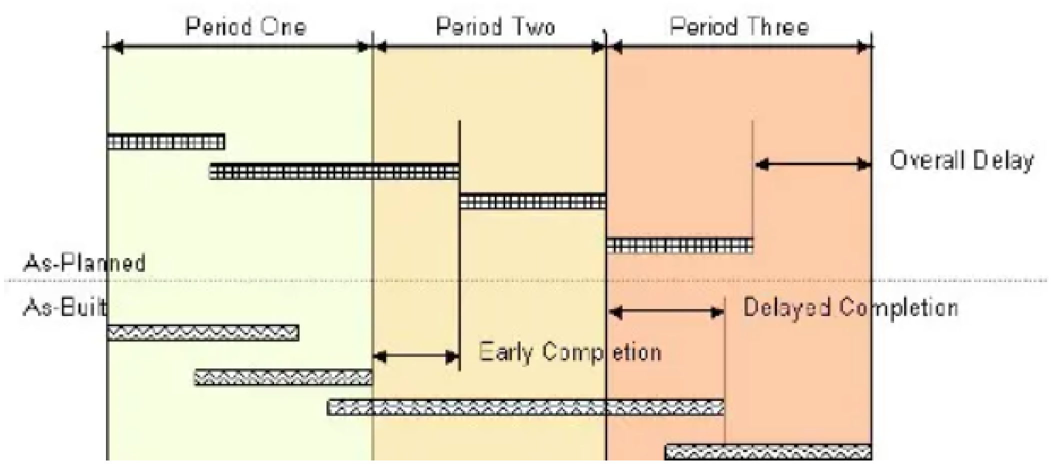
As-planned vs. as-built windows analysis – windows analysis

- When you have **adequate project record documentation**
  - Here you can realize if **critical paths shifted** at least at **key dates!**
    - ✓ **Contemporaneous critical path**



As-planned vs. as-built windows analysis – windows analysis

- When you have **adequate project record documentation**
  - Here you can realize if **critical paths shifted** at least at **key dates**
    - ✓ **Contemporaneous critical path**



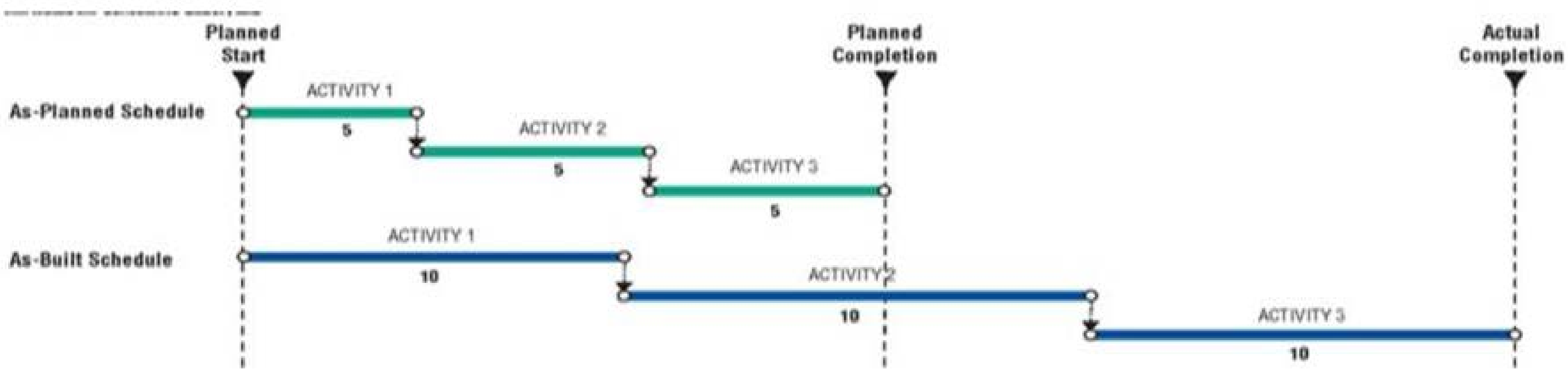
Source: AACE International Recommended Practice No 29R-03 – Forensic Schedule Analysis, AACE International, Morgantown, WV, 2011

What is the difference between
as-planned vs. as-built windows analysis
And
impacted as-planned analysis?

What is the difference between
as-planned vs. as-built windows analysis
And
time slice windows analysis?

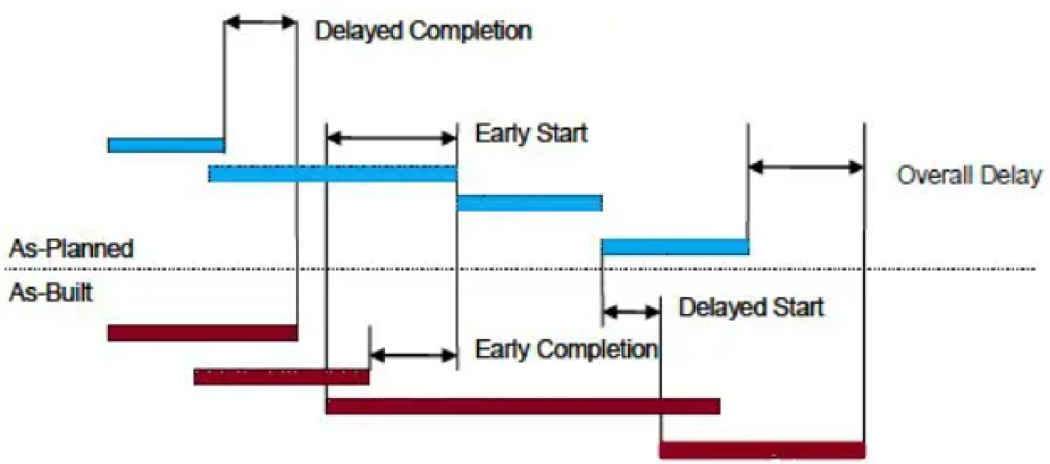
As-planned vs. as-built windows analysis – Total time analysis

- When you have **inadequate project record documentation**
  - Here you **cannot** realize if **critical paths shifted**
    - ✓ **Retrospective (As-built) critical path**



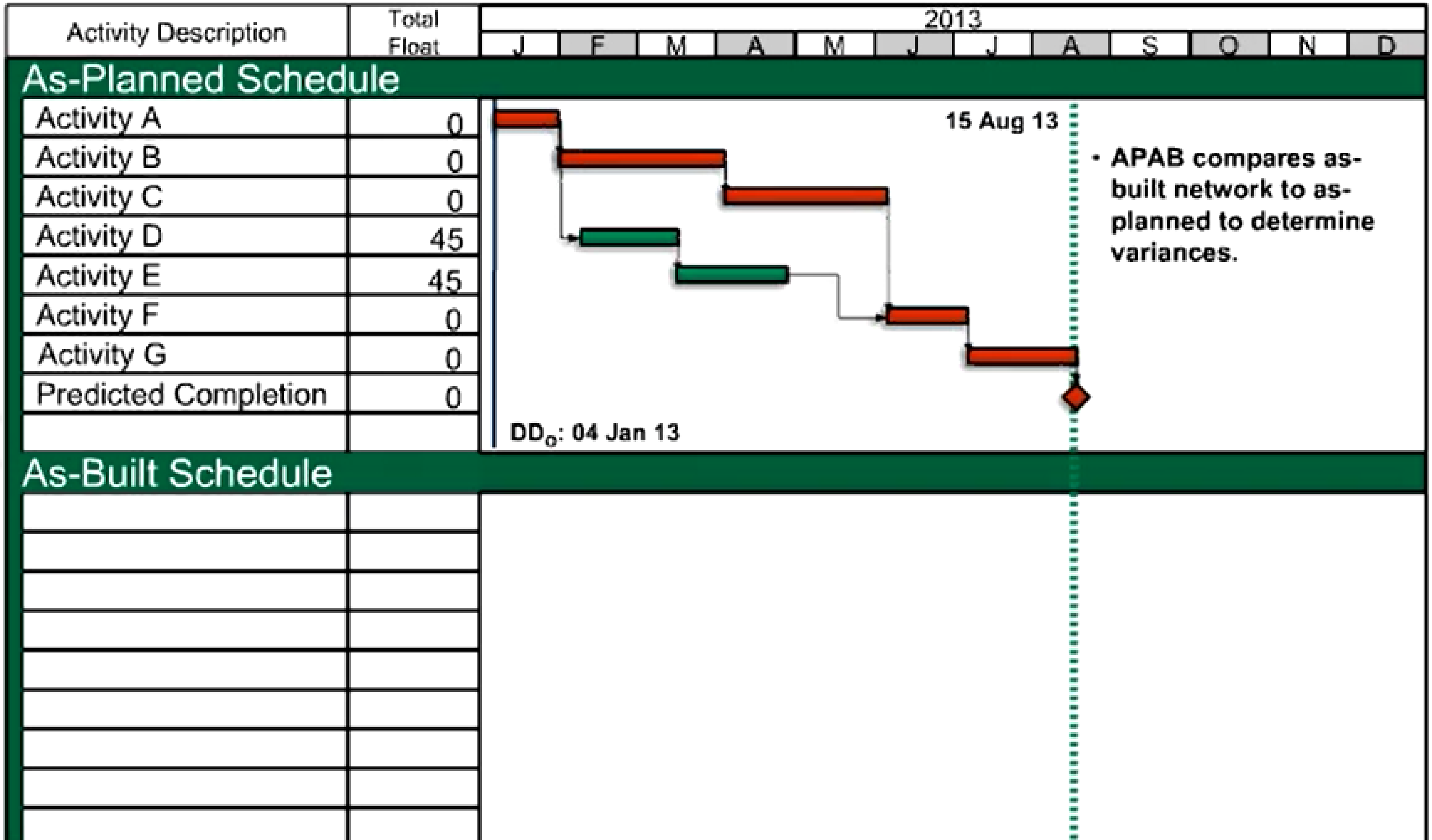
As-planned vs. as-built windows analysis – Total time analysis

- When you have **inadequate** **project record documentation**
  - Here you **cannot** realize if **critical paths shifted**
    - ✓ **Retrospective (As-built) critical path**

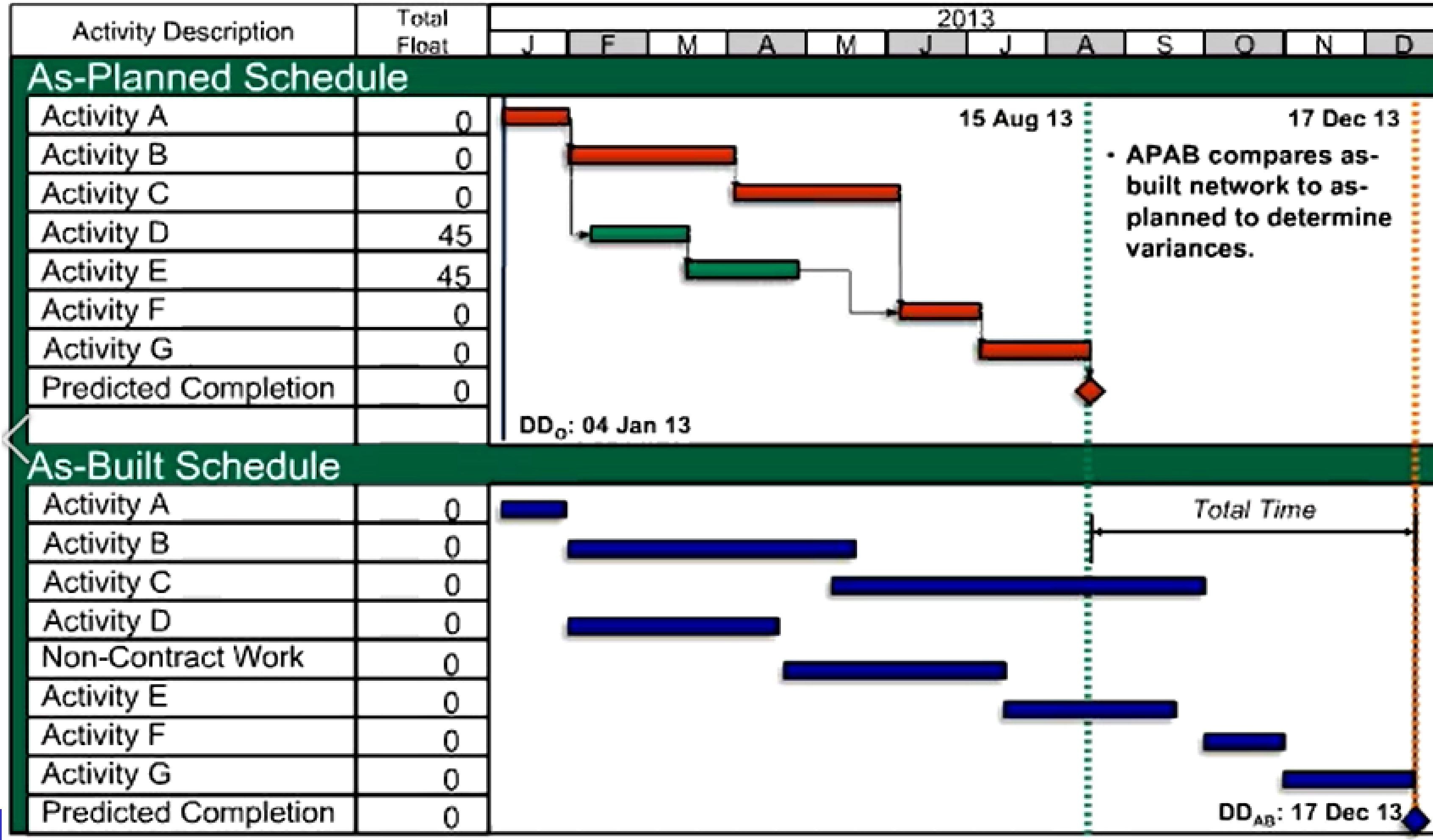


Source: AACE International Recommended Practice No 29R-03 – Forensic Schedule Analysis, AACE International, Morgantown, WV, 2011

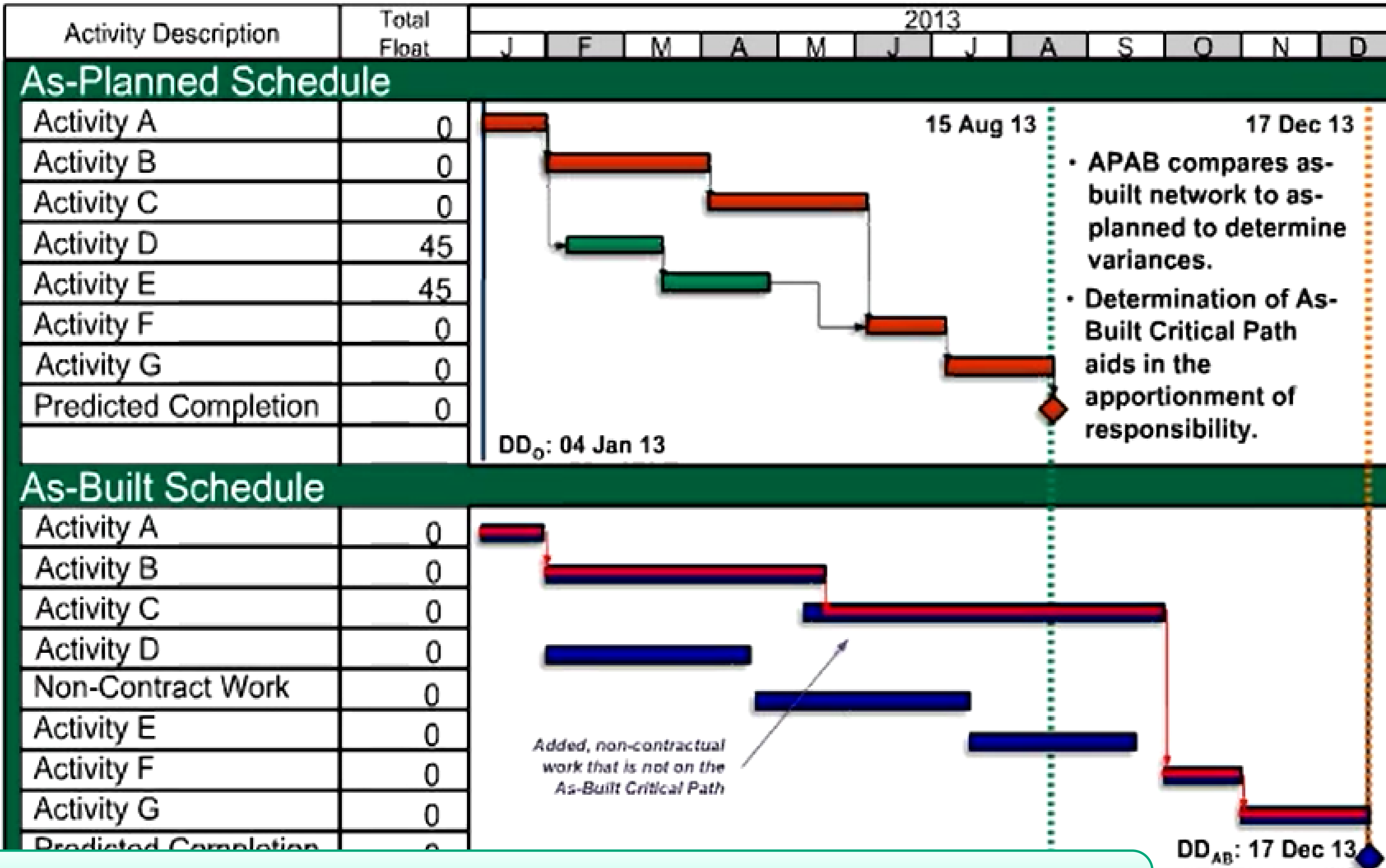
As-planned vs. as-built windows analysis – Total time analysis Example



As-planned vs. as-built windows analysis – Total time analysis Example



As-planned vs. as-built windows analysis – Total time analysis Example

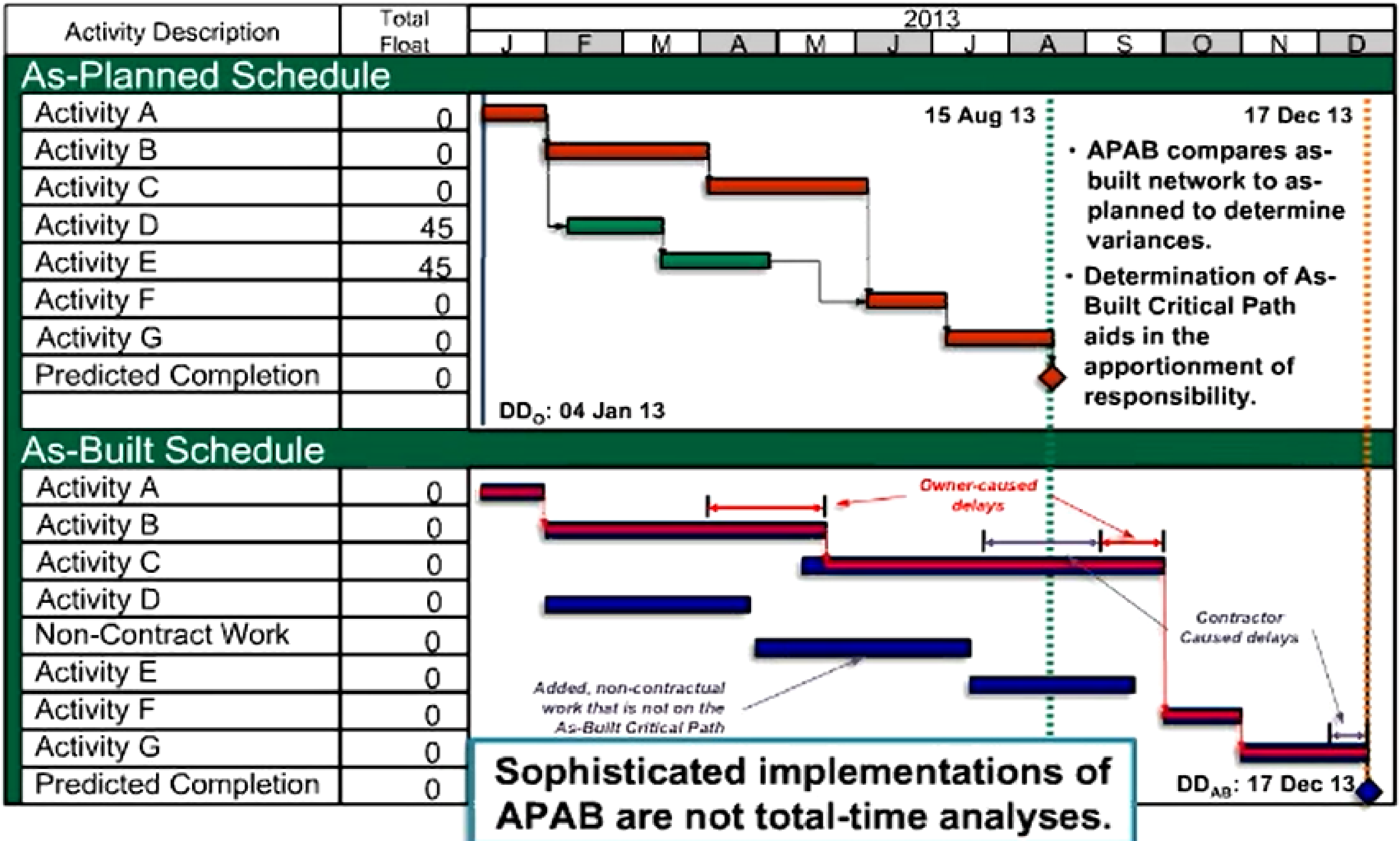


Caution:

Do not use total time analysis in this case == The court will reject

ذکر مطالب فوق تنها

As-planned vs. as-built windows analysis – Total time analysis Example



As-planned vs. as-built windows analysis – Pros and cons

If windows analysis is used:

- ✓ **Low to Moderate** in terms of **cost** and **time**
  - ✓ However, if the **as-built data** are **not accurate**, this can be **very expensive**
- ✓ **Contemporaneous critical path** if enough records are available.
- ✓ **Used** for **after-the-fact** schedule (**retrospective delay impact**)
- ✓ It **considers actual progress** and **changes (not hypothetical)**
- ✓ **Accepted** by **many courts** if performed as windows basis
  - × Rejected by many if total analysis used

As-planned vs. as-built windows analysis – Pros and cons

- **As-built sequence** must relate to **as-planned sequence** for activity **level variance method**
  - If the **sequences of activities** have a **great variance**, it will have **lower accuracy**
- **Concurrent delays** may not be considered if don’t have enough records
- **May** be **Subjective**
  - It **can rely** heavily on the **rationale** and **determination** of **analyst**
  - **Susceptible** to **unintentional** or **intentional manipulation** by choice of as-built data that is **incorporated** into **schedule**
- **Can** be **difficult** if the project is **complex** and **long**
  - Since it does **not rely** on **programming software**

As-planned vs. as-built windows analysis – When to use?

- It is **usually** applied **when**:

1. There is **concern** over the **validity** or **reasonableness** of the **baseline schedule**
2. There is **concern** over the **validity** or **reasonableness** **contemporaneously updated schedules**
3. There are **too few** **contemporaneously updated schedules**.
4. There are **baseline schedule (only Gantt chart)** + **As-built data**

When the benefits of this approach decrease?

As-planned vs. as-built windows analysis – When to use?

Caution:

The benefits of this approach are decreased when:

1. **Regular updates are available**
2. **Not well-suited to determining delay causation when a project is complex and spans longer durations**
3. **Not suitable for projects built in a manner significantly different than planned**
   - ❖ **This has lot to do with discretionary logic === we will talk later**
4. **Not typically appropriate for projects that include near-critical paths that experienced delay,**

   as the method cannot appropriately consider shifts in the critical path over time.
   - ✓ **Before using this method, the float analysis should be performed**

As-planned vs. as-built windows analysis – When to use?

The main drawback is that the costs associated with the delay in each window would require an average prolongation rate if the duration of each window increases

When the duration of each window increases == average rate is more applicable rather than calculating actual cost

As-planned vs. as-built windows analysis – Practical procedure

1. **The windows should be selected first.**

- The beginning and end of each "window" is usually defined by:
  - **Key milestone events** which occurred during the course of the works.
  - **Periods** when the contractor's **intentions**, **logic** or **methodology** **changed** (e.g. **attempted acceleration or changed conditions**).
  - **Periods** when snapshots of progress were recorded contemporaneously, allowing **earned value analysis calculations** **within each window**.
  - **Key changes**, **suspensions** or **delays** having been experienced.

As-built critical path windows.

Window Ref.	Description	Actual Start	Actual Finish
CP-1.001	Install pipe	6-Jan-06	12-Feb-06
CP-1.002	Test pipe	13-Feb-06	30-May-06
CP-1.003	Handover	31-May-06	28-Jun-06

As-planned vs. as-built windows analysis – Practical procedure

2. Window level variances reconciliation table (WLVs) should be calculated

Window level variance reconciliation table.

Window Ref.	Planned Start	Planned Finish	Actual Start	Actual Finish	Start var. (less prev. WLV)	Finish var. (less prev. WLV)	WLV
CP1-001	1-Jan-06	7-Feb-06	6-Jan-06	12-Feb-06	5	–	5
CP1-002	8-Feb-06	12-Feb-06	13-Feb-06	30-May-06	–	102	102
CP1-003	13-Feb-06	28-Feb-06	31-May-06	28-June-06	–	13	13

As-planned vs. as-built windows analysis – Practical procedure

3. Liability assessment

- To allow **apportionment** of the delay in each window, the WLV analysis requires one **additional step of liability assessment**.

Liability assessment.

Window Ref. CP	Description	WLV	CDE	EDE	Liability Assessment
CP1-001	Install pipe	5		5	Delayed site access (5 days)
CP1-002	Test pipe	102	82	20	Pipe repair (80 days), late instruction (20 d), plant breakdown (2 days)
CP1-003	Handover	13		13	Additional back-fill (10 days), additional manhole (3 days)

After-the-Fact Schedule Analysis

(Retrospective longest path analysis (RLPA))

FSA methods

Method of Analysis	Analysis Type	Critical Path Determined	Delay Impact Determined	Requires
Time Slice Windows Analysis	Effect & Cause	Contemporaneously	Retrospectively	• Logic linked baseline programme. • Update programmes or progress information with which to update the baseline programme.
As-Planned versus As-Built Windows Analysis	Effect & Cause	Contemporaneously	Retrospectively	• Baseline programme. • As-built data.
Retrospective Longest Path Analysis	Effect & Cause	Retrospectively	Retrospectively	• Baseline Programme. • As-built programme.
Collapsed As-Built Analysis	Cause & Effect	Retrospectively	Retrospectively	• Logic linked as-built programme. • A selection of delay events to be modelled.

FSA methods

The Information Required					
	Baseline programme	Logic linked baseline programme	As-built data	Delay events to model	Up to date programmes/ process information
Time Slice Windows					
As-Planned v As-Built Windows					
Longest Path					
Collapsed As-Built					

The table above shows the evidence required for each delay analysis method. The table has been simplified.

Retrospective longest path analysis – Definition

- It involves the determination of the **"retrospective as-built critical path."**
- It **adopts** "the perspective evident at the **end of the project**."
- The **prerequisite** for this type of analysis is:
  1. A **baseline** schedule
  2. An **as-built schedule** and/or **records**.

Retrospective longest path analysis – Definition

- In this method the analyst **must first** either **verify an existing as-built schedule**

What if an as-built schedule is not available?

Develop a detailed one

The **as-built schedule** does **not need** to be at the **same level of detail** as the **baseline schedule**, **unlike** the **As-planned vs As-Built Windows Analysis**.

Retrospective longest path analysis – How to use?

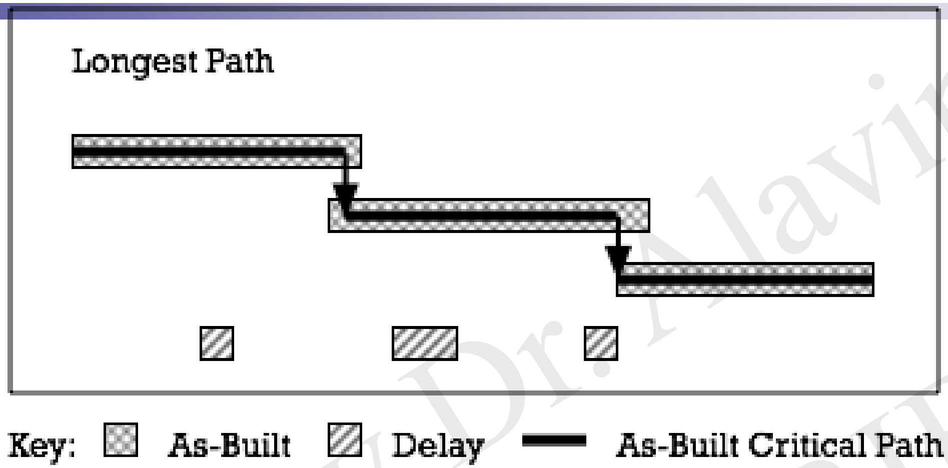
1. Develop **as-built schedule**
2. **Analyst traces** the **longest continuous path backwards** from the **actual completion** date to determine the **as-built critical path**
3. **key dates compared** along
   - The **retrospective (as-built) critical path**

     against

   - Corresponding **planned dates** in the **baseline schedule**.

4. Thereafter, the **analyst investigates** the **project records** to **determine** what delay **events might have caused** the **identified critical delay**

Retrospective longest path analysis – Example



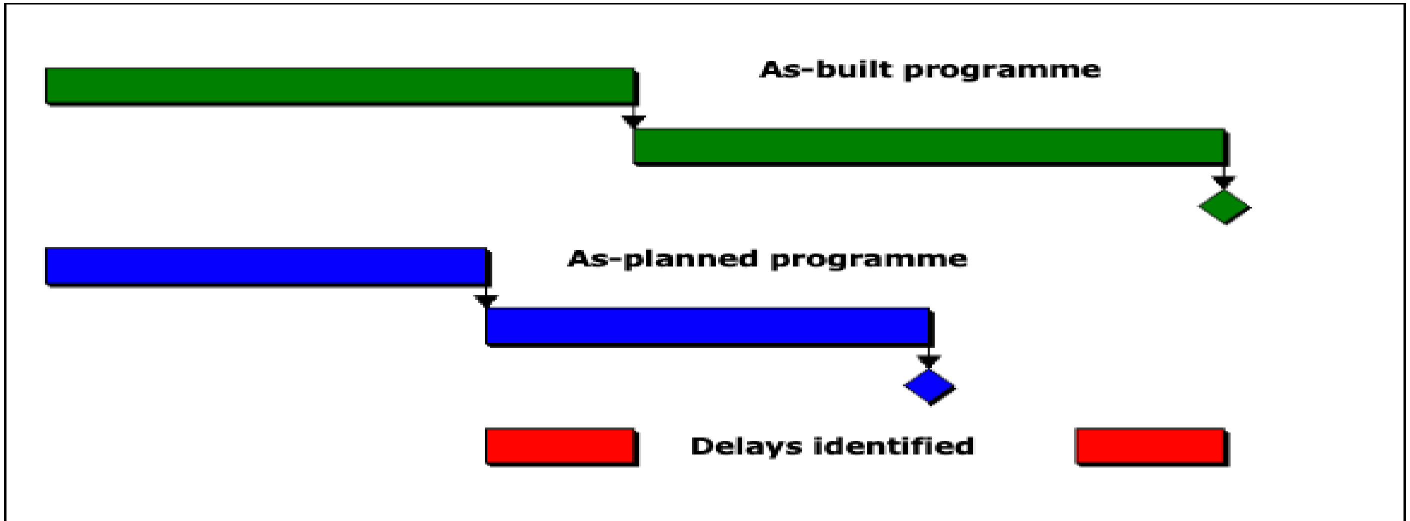


Diagram illustrating the Longest Path technique.

Have you realized something?

Retrospective longest path analysis – Pros and cons

- ✓ **Low** in terms of **cost** and **time**
- ✓ **Can** be used for **after-the-fact** schedule (**retrospective delay impact**)
- × **Static analysis** (**Retrospective critical path**)
  - × **Limited capacity** to **recognize** and **allow** for **switches** in the **critical path**
- × It **may not consider actual progress** and **changes (may be hypothetical)**
- × **Subjective**
  - × In terms of **actual progress** and **changes**
  - × In terms of **critical path**
- × **May not allow** analyst to **identify** and **quantify every day of delay**
- × Cannot consider **concurrent delays**

Retrospective longest path analysis – When to use?

- It is **usually** applied **when** there are **<u>no</u> update schedules** or **usable update schedules**
- There are **baseline schedule (only Gantt chart)** + **As-built data**

Caution:
It is recommended to use other methods to support this analysis

e.g., resource curve analysis or earned value analysis

After-the-Fact Schedule Analysis

(Collapsed As-Built Analysis (CAB))

FSA methods

Method of Analysis	Analysis Type	Critical Path Determined	Delay Impact Determined	Requires
Time Slice Windows Analysis	Effect & Cause	Contemporaneously	Retrospectively	• Logic linked baseline programme. • Update programmes or progress information with which to update the baseline programme.
As-Planned versus As-Built Windows Analysis	Effect & Cause	Contemporaneously	Retrospectively	• Baseline programme. • As-built data.
Retrospective Longest Path Analysis	Effect & Cause	Retrospectively	Retrospectively	• Baseline Programme. • As-built programme.
Collapsed As-Built Analysis	Cause & Effect	Retrospectively	Retrospectively	• Logic linked as-built programme. • A selection of delay events to be modelled.

FSA methods

The Information Required

	Baseline programme	Logic linked baseline programme	As-built data	Delay events to model	Up to date programmes/ process information
Time Slice Windows					
As-Planned v As-Built Windows					
Longest Path					
Collapsed As-Built					

The table above shows the evidence required for each delay analysis method. The table has been simplified.

Collapsed as-built analysis – Definition

- SCL Protocol:
  - Very **similar** to **impacted as-planned** (**in reverse**)
  - Also known as "**As-Built But for (ABBF)**" analysis
  - This is a **what if scenario**
  - It involves the **extraction of delay events** from the **as-built schedule** to provide a **hypothesis** of what **might have happened** had the **delay events not occurred**.
  - It does **not require** a **baseline schedule**.
  - It requires a **detailed logic-linked as-built schedule**.
    - ✓ It is **rare** that such a schedule would **exist** on the project
      - Therefore the analyst is usually required to **introduce logic** to a **verified as-built schedule**.
        - This can be **very time consuming** and **complex endeavor**.

Collapsed as-built analysis – Definition

- SCL Protocol:
  - Once **completed (logic-linked as-built created),**

    sub-networks for the delay events within the as-built programme are identified

    they are **'collapsed' or extracted** in order to determine the **net impact** of the delay events.

  - This method is sometimes done in **windows**, using interim or **contemporaneous programmes** which contain detailed and comprehensive as-built data.

Collapsed as-built analysis – Definition

- SCL Protocol says:

'4.7 Collapsed as-built is based on the effect of employer risk events on the programme of work as it was actually built. Similar to the as-planned versus as-built, the use of this technique is restricted by its inability to identify concurrency, resequencing, redistribution of resources or acceleration. This is particularly the case when the nature of the as-built logic is complex, requiring subjective reconstruction of as-built logic. Where acceleration, redistribution of resources or resequencing has taken place during the course of the works to overcome the effects of events, this form of analysis may produce unreliable results'.

- The SCL Protocol is **justifiably cautious** about recommending the **CAB approach.**

Collapsed as-built analysis – Definition

- AACE **method implementation protocol (MIP) 3.8 and 3.9** say:
  - **Recreates a CPM model** starting with the **as-built schedule**, then **deletes selected delay activities** in order to **show** what **would have happened**, had that **event not taken place**

Collapsed as-built analysis – How to use?

NTP = NET TIME PRESENT
AS=ACTUAL START
ABCP=ACTUAL BASED CRITICAL PATH

Collapsible As-Built

- Schedule created with a Data Date equal to NTP, original durations equal to actual durations, and logic designed to make ES/EF equal to AS/AF.
- Also contains activities representing delays.
- ABCP is calculated.

Collapsed As-Built

- Delay activities are dissolved, in reverse order.
- Schedule is recalculated after each dissolution.
- Method purports to demonstrate what would have happened "but for" the dissolved events.

Collapsed as-built analysis – How to use?

1. Develop a **detailed logic-linked as-built schedule**.
2. **Sub-networks** (fragnets) for the **delay events** are **identified** within **as-built schedule**
3. **Collapse** and **extract Sub-networks** (fragnets) for the **delay events** in order to determine the **net impact** of the **delay events**

- This method is **sometimes** done in **windows**, using **interim** or **contemporaneous schedules** which contain **detailed** and **comprehensive as-built data**.

Before any collapsing can take place, the delay events relied on EDE, CDE, or both, must be discretely identified among the as-built activities.

These can be in the form of "fragnets."

Collapsed as-built analysis – Example

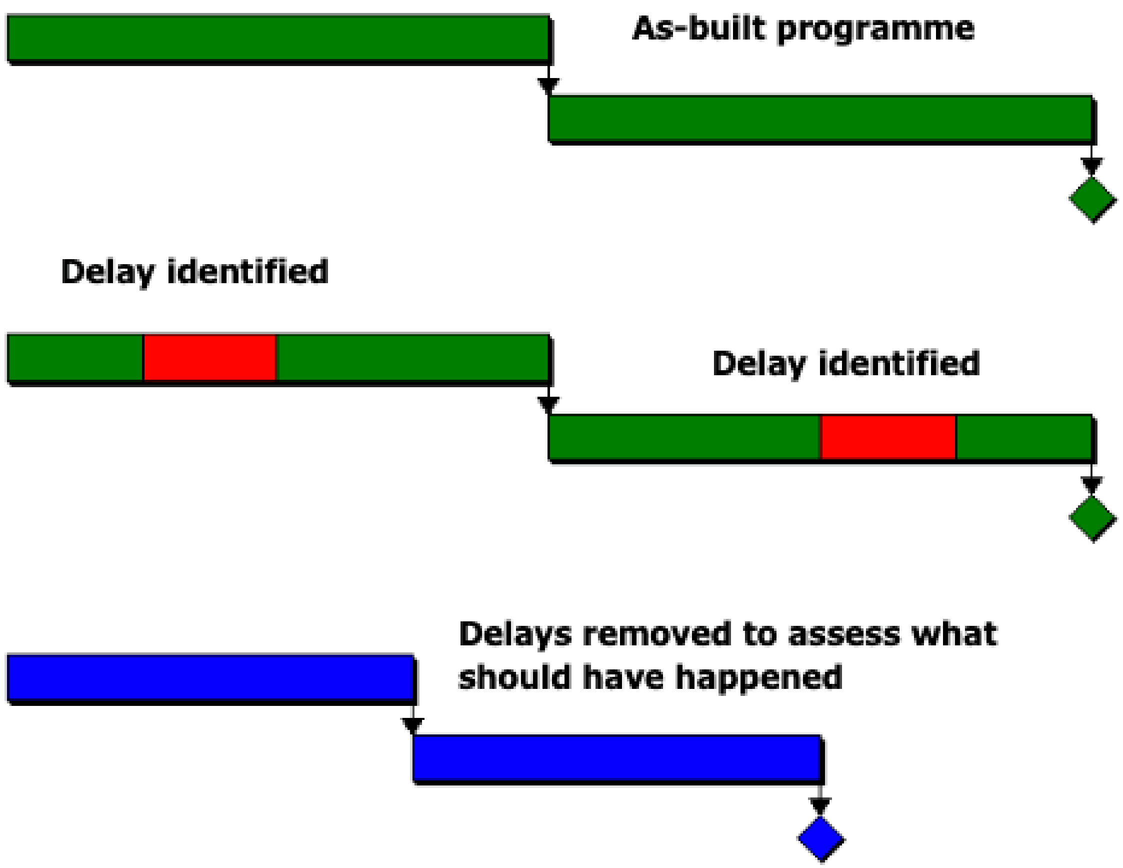
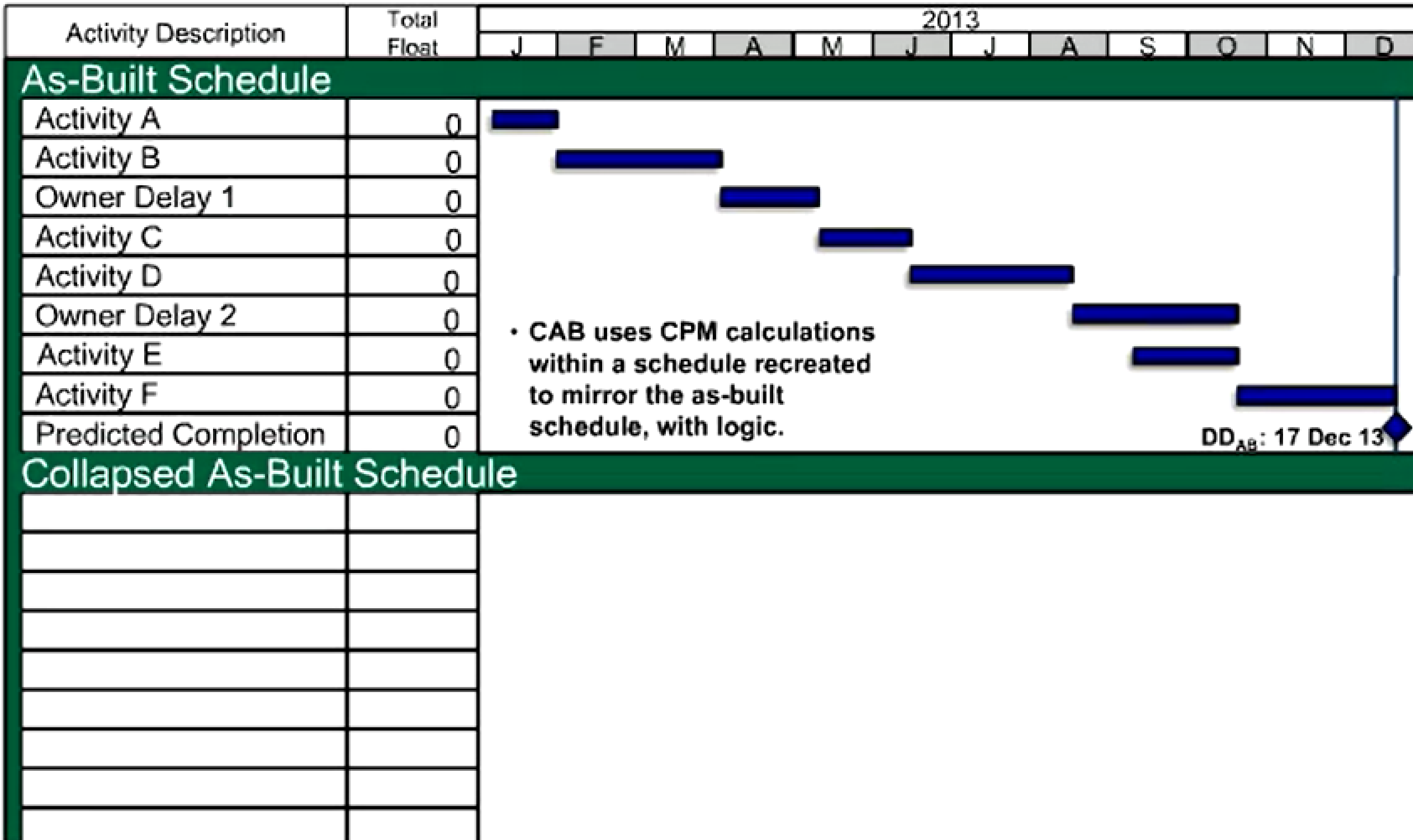
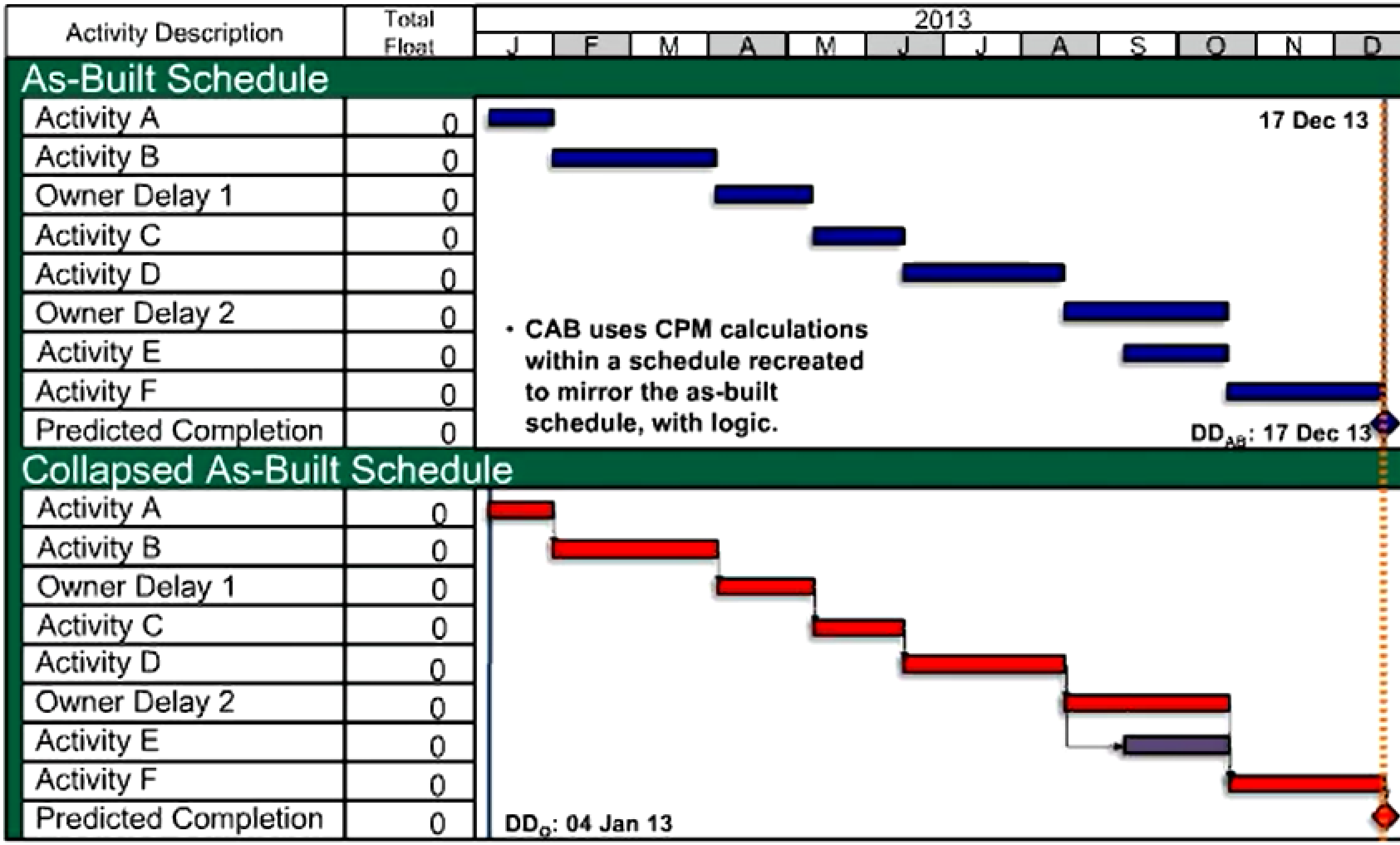


Diagram illustrating the Collapsed As-Built technique.

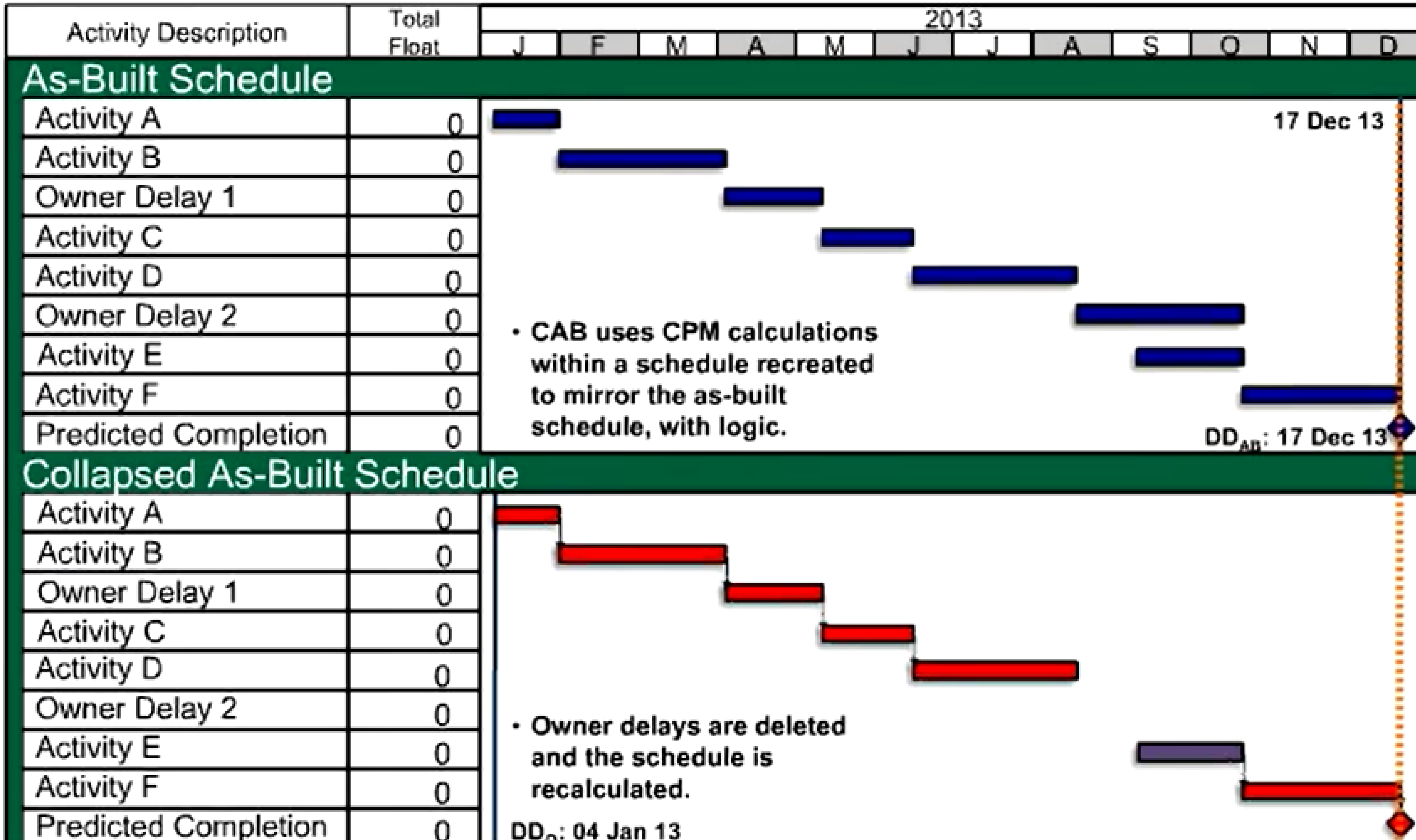
Collapsed as-built analysis – Example



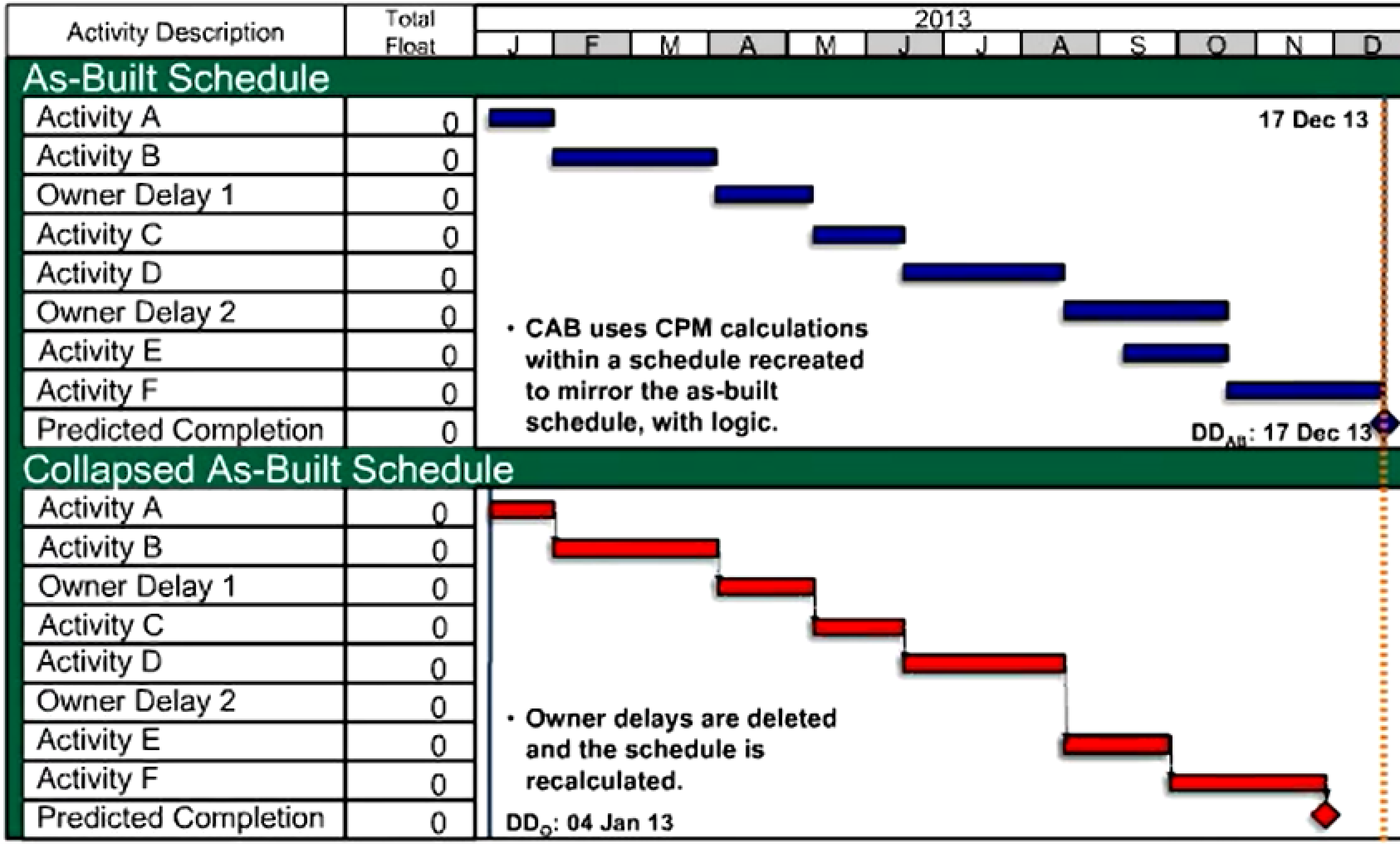
Collapsed as-built analysis – Example



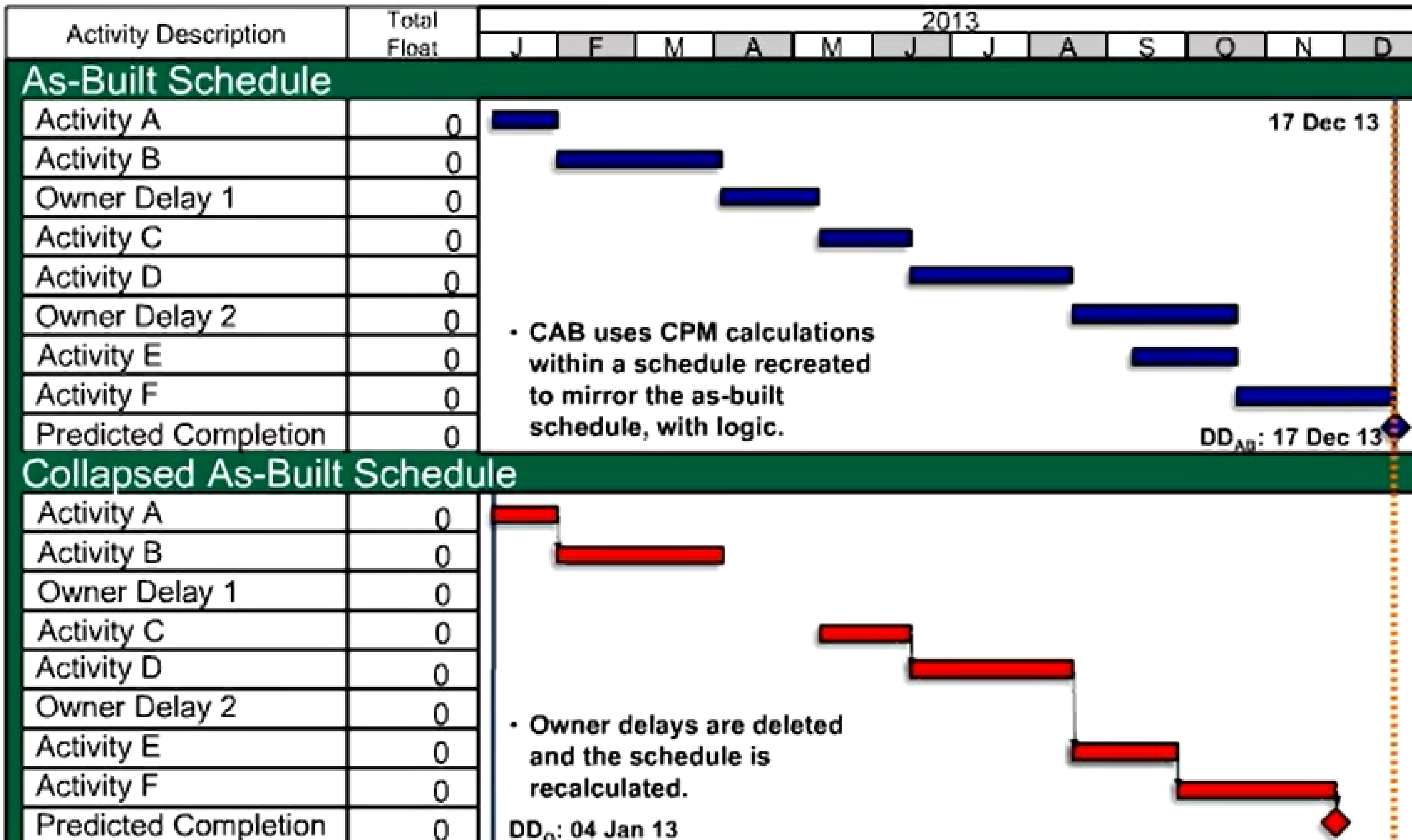
Collapsed as-built analysis – Example



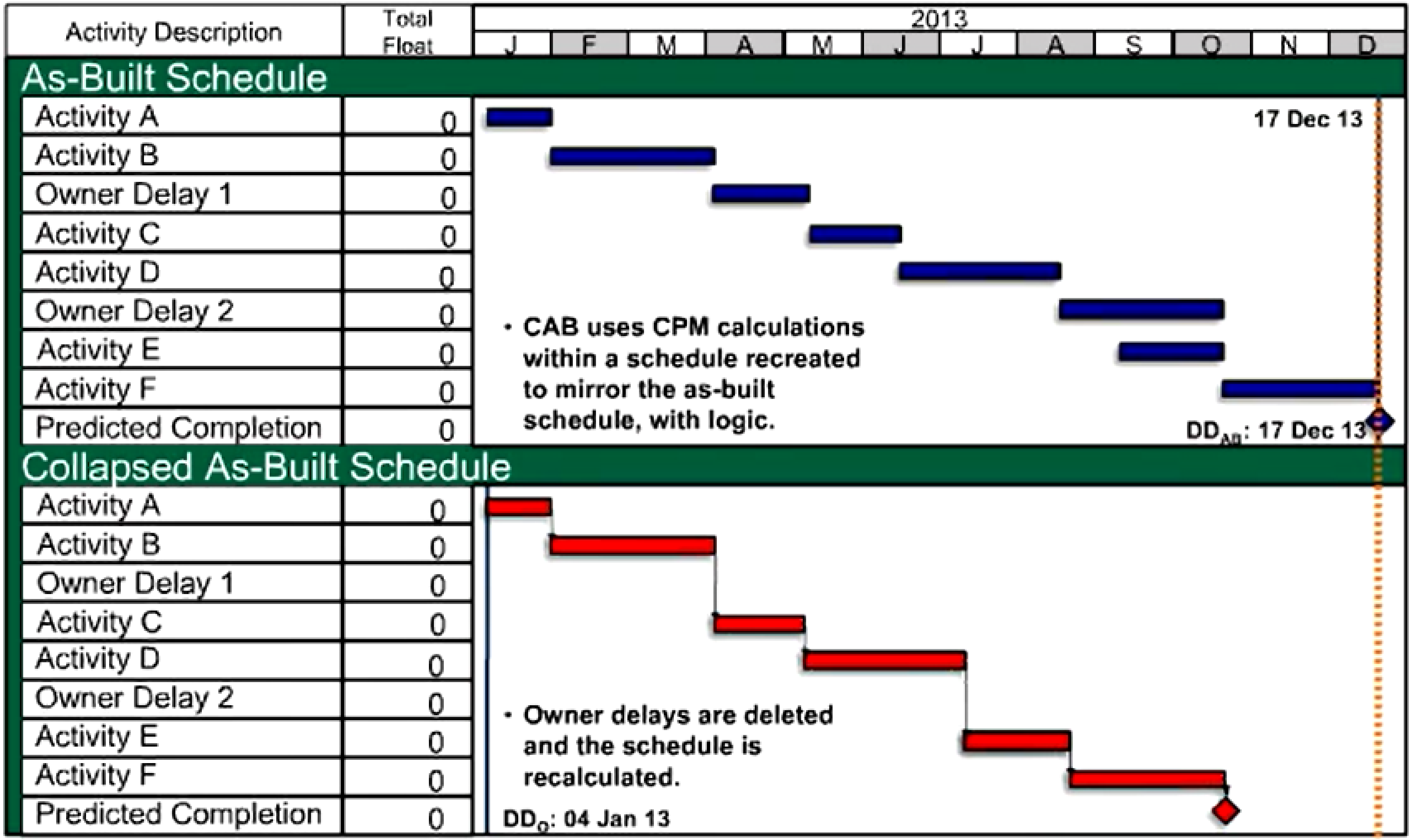
Collapsed as-built analysis – Example



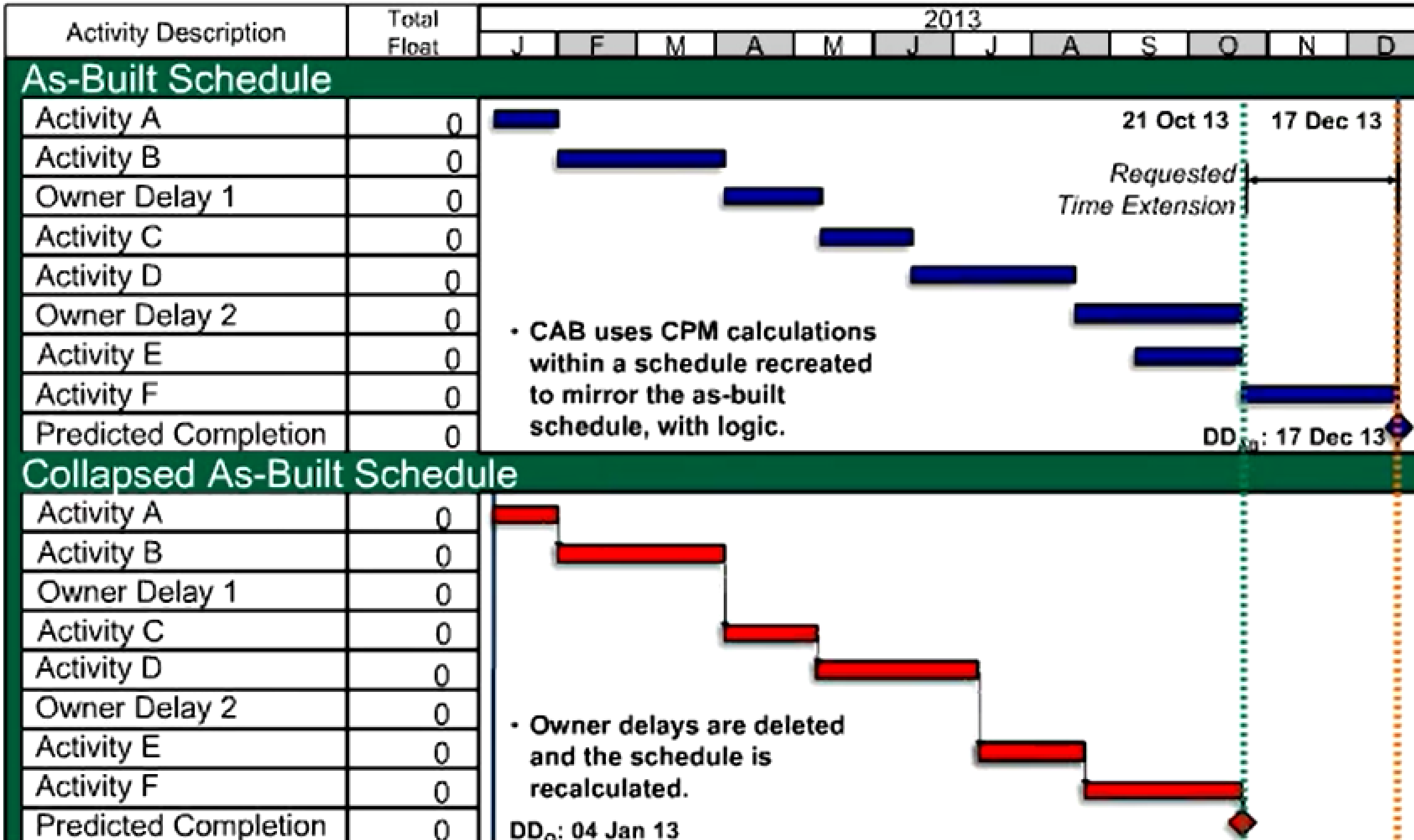
Collapsed as-built analysis – Example



Collapsed as-built analysis – Example



Collapsed as-built analysis – Example



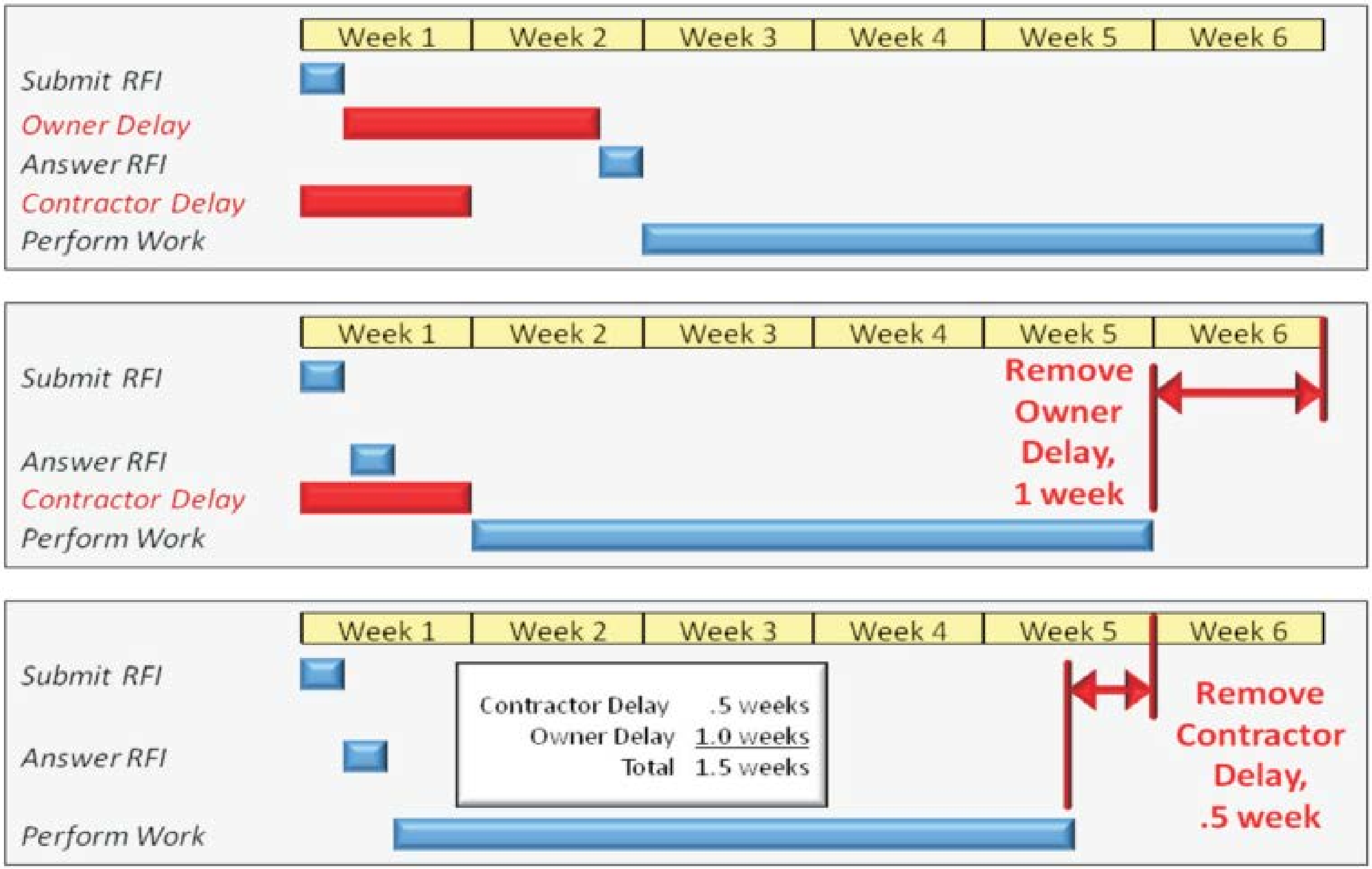
IS IT ALWAYS THAT SIMPLE?

Collapsed as-built analysis – Example

As-Built Schedule with Delays

As-Built Schedule with Owner Delay Removed First

As-Built Schedule with Contractor Delay Then Removed

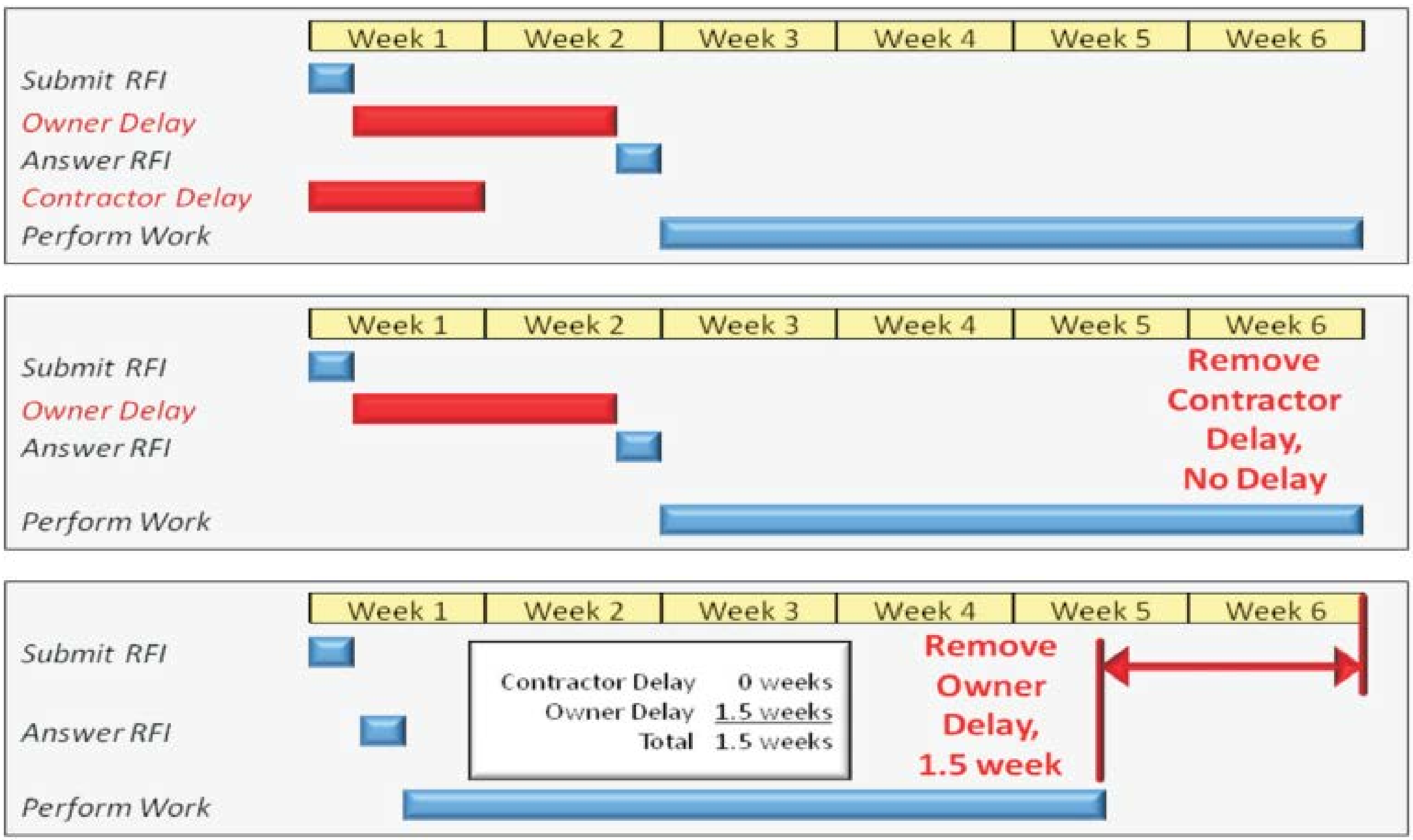


Collapsed as-built analysis – Example

As-Built Schedule with Delays

As-Built Schedule with Contractor Delay Removed First

As-Built Schedule with Owner Delay Then Removed



Collapsed as-built analysis – Pros and cons

- ✓ **Often known as "but-for" analysis**
- ✓ **Easy to understand**
- ✓ Does not require **baseline programme**
- ✓ Can isolate impact of EDE from CDE when **iterative applications** applied
- ✓ Used for **after-the-fact** schedule (**retrospective delay impact**)

Collapsed as-built analysis – Pros and cons

- × **Rejected by many courts <u>especially</u>** if the as-built schedule is not well developed
- × **Static analysis** (**Retrospective critical path**)
  - × Collapsed As-Built Analysis **“freezes” the critical path**.
  - ✓ It could be improved **if step-based approach and window analysis** used.
    - × The critical path then **changes depending on the “subtraction” made**, but these critical path “shifts” **rarely match** the shifts shown in the **<u>contemporaneous</u> project schedules**.
- × **Subjective** in terms of **actual progress** and **changes** (using all **new logic and progress**)
- × **Ignores plan** because you heavily **rely** on **as-built**
- × **Can** be extremely **complicated**
- × **Susceptible** to **manipulation** during **as-built logic assignments**
- × **Moderate** to **very high** in terms of **time** and **cost**
- × **No consideration of concurrent delays** if no step-based approach is used.

Collapsed as-built analysis – When to use?

- The **actual effect** of the **delay** event is **known**.
- An **accurate as-built schedule** is available or can be **developed**.
- A **baseline schedule** is **not available**.

Before using this method, the analyst needs to ask some questions!

موسسه مهندسی و مدیریت ساخت علوی پور (ACEMI)

دفتر آموزش
تهران، بلوار فردوس شرق، مجتمع آبگینه، طبقه سوم اداری، واحد C42

+۹۸ ۲۱ ۴۶ ۱۰۰ ۴۴۵
+۹۸ ۲۱ ۴۶ ۱۰۰ ۴۵۰

info@dralavipour.com
www.dralavipour.com
@alavipour_pm
@acemi_social
@alavipour_pm
@IRANACEMI

ACEMI