

# Concurrent Delay의 식별

2020.06

임정주 대표 (The TEAM)

# 1 Concurrent Delay 식별

## 1) Delay Analysis Method in SCL Protocol

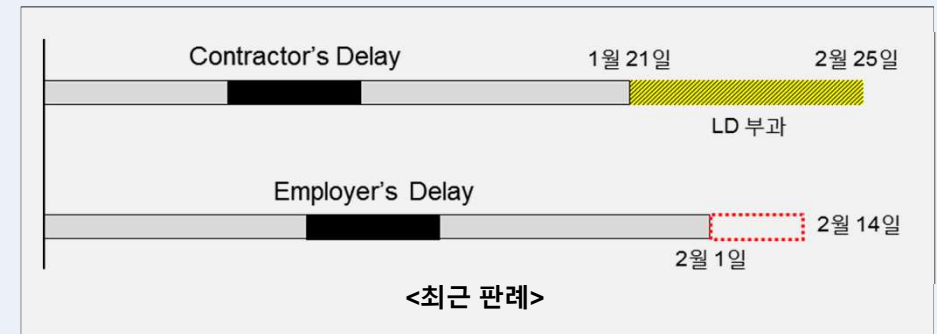
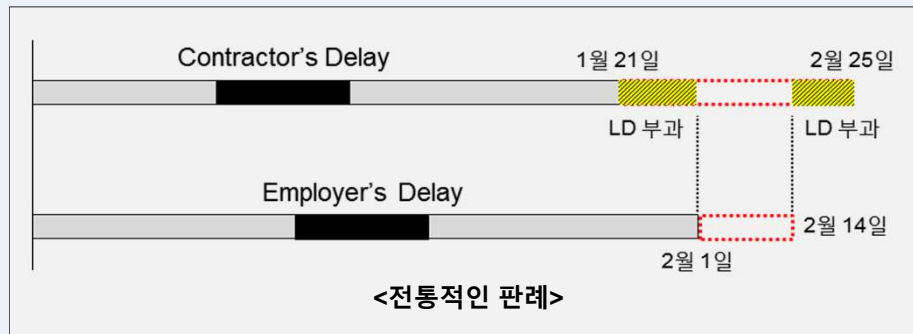
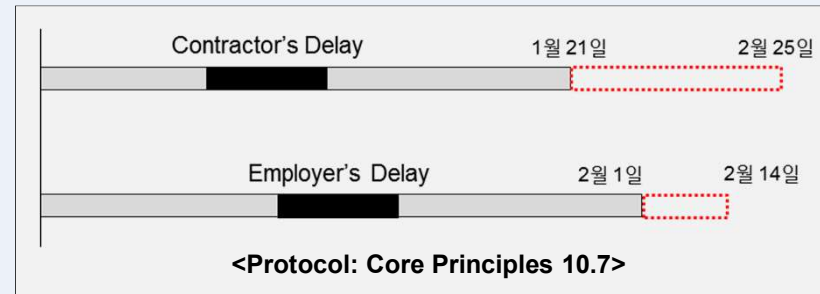
Method	Analysis Type	Critical Path Determined	Delay Impact Determined	내용
Impacted As-Planned Analysis	Cause & Effect	Prospectively	Prospectively	<ul style="list-style-type: none"> <li>Baseline과 Fragnet으로 분석하는 방법이며, 사용하기에 쉽지만 Actual을 고려하지 않기 때문에 합리적이지 않음.</li> </ul>
Time Impact Analysis		Contemporaneously		<ul style="list-style-type: none"> <li>Updated Program과 Fragnet을 필요로 하며, 가장 합리적인 분석방법으로 인정 받고 있음.</li> </ul>
Time Slice Window Analysis	Effect & Cause	Contemporaneously	Retrospectively	<ul style="list-style-type: none"> <li>Updated Program을 필요로 하며, Window Analysis 방법 중 하나이며, 월별로 CP를 확인하는 방법임.</li> </ul>
As-Planned v As-Built Window Analysis				<ul style="list-style-type: none"> <li>Baseline, As-built data를 필요로 하며, Window Analysis 방법 중 하나이며, 프로그램을 꼭 써야 하는 것은 아님.</li> </ul>
Retrospective Longest Path Analysis		Retrospectively		<ul style="list-style-type: none"> <li>Baseline과 As-Built Program을 필요로 하며 Longest Path를 추적하여 확인함. CP의 변동성을 인식하기는 어려움.</li> </ul>
Collapsed As-Built Analysis	Cause & Effect	Retrospectively		<ul style="list-style-type: none"> <li>As-Built Program, Fragnet을 필요로 하며, Updated Program이 있다면 시간의 반대순으로 진행하는 방법임.</li> </ul>

# 1 Concurrent Delay 식별

## 2) Concurrent Delay in SCL Protocol

"Where Contractor Delay to Completion occurs or has an effect concurrently with Employer Delay to Completion, **the Contractor's concurrent delay should not reduce any EOT**"

The Society of Construction Law, Delay and Disruption Protocol 2<sup>nd</sup> Edition, 2017



- SCL Protocol은 최근 판례를 고려한 우측 Case를 추천하고 있음.

# 1 Concurrent Delay 식별

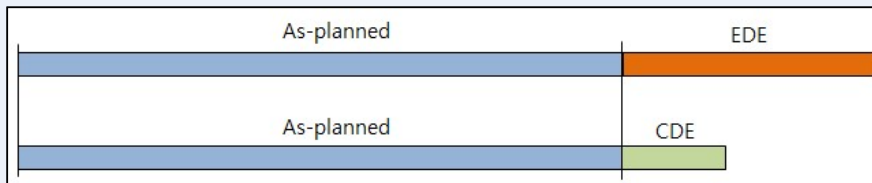
## 3) Impacted As-Planned

### (1) Combined Chronological Impacted As-Planned (CC-IAP)

Combined impacted liability table			Cumulative delay	
Event	Event type	Impacted completion date	EDE	CDE
Baseline		1-Jan-09		
001	EDE	2-Jan-09	1	
002	EDE	5-Jan-09	3	
003	CDE	8-Jan-09		3
004	EDE	10-Jan-09	2	
005	CDE	15-Jan-09		5
006	CDE	15-Jan-09		0
007	EDE	18-Jan-09	3	

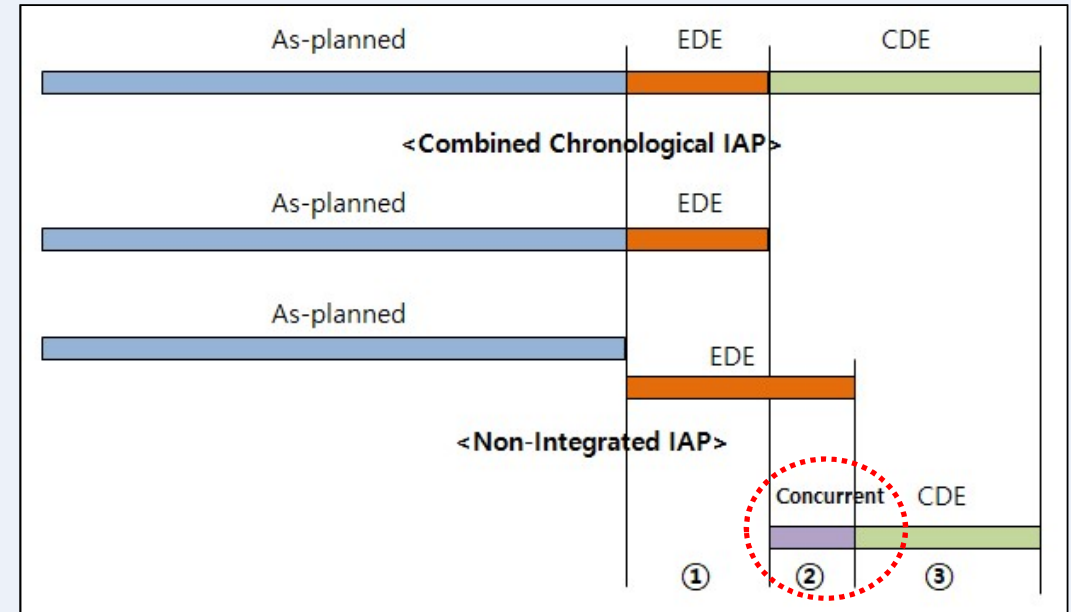
- Baseline에 EDE와 CDE를 넣어서 지연기간을 계산하는 방법임.
- 발생한 순서에 따라 입력하고 지연기간을 각각 계산해서 더함.

### (2) Non-Integrated Impacted As-Planned (NI-IAP)



- Baseline에 EDE와 CDE를 넣는 것은 동일하지만 각각 입력하여 2개의 프로그램이 생성되며 각각 지연기간을 계산함.

### (3) Concurrent Delay의 식별



- CC-IAP에서 CDE를 빼면 순수한 EDE에 의한 기간 ①이 산출되며, 이 기간은 Compensable Delay기간에 해당됨.
- ①기간은 NI-IAP의 EDE기간보다 짧아지는데, 그 차이가 ②번이며 Approximate Concurrency로 인식될 수 있음.
- ③기간은 Non-excusable Delay로 LD가 부과되는 기간임.

# 1 Concurrent Delay 식별

## 4) Time Impact Analysis

Event (fragnet)	Event type	Event actual start date	Base schedule	Base schedule data date	Projected completion date	Net loss/gain	Cumulative delay		
							EDE	CDE	Concurrent
001	EDE	3-Jun-08	Baseline	1-Jun-08	1-Jan-09	1	1		
002	EDE	5-Jun-08		1-Jun-08	5-Jan-09	3	3		
			UD01	30-Jun-08	5-Jan-09	0			
003	CDE	2-Jul-08		30-Jun-08	8-Jan-09	3			3
004	EDE	2-Jul-08		30-Jun-08	10-Jan-09	2	2		
			UD02	31-Jul-08	6-Jan-09	-4			-4
			UD02A	14-Aug-08	8-Jan-09	2			2
005	CDE	15-Aug-08		31-Jul-08	15-Jan-09	7			7
			UD03	31-Aug-08	14-Jan-09	-1			-1
006	CDE	4-Sep-08		31-Aug-08	15-Jan-09	1			1
007	EDE	7-Sep-08		31-Aug-08	18-Jan-09	3	3		
<b>Totals:</b>						<b>17</b>	<b>9</b>	<b>4</b>	<b>4</b>

- 003 CDE Event는 Concurrent Delay로 계산되었음. 004 EDE Event와 동시에 시작되었고 EDE는 5일의 영향을 발생시켰기 때문에 겹치는 기간인 3일만 Concurrent Delay로 계산되고, 추가적인 2일은 Employer가 책임지는 기간으로 계산되었음.
- 006 CDE Event는 Concurrent Delay로 계산되었음. 007 EDE Event가 4일의 영향을 발생시켰기 때문에 겹치는 1일이 Concurrent Delay로 계산되었음. (그러나 이런 계산은 Event의 발생시점을 고려하였을 때 적절하지는 않은 것으로 판단됨)
- 분석 전에 Event 성격을 Concurrent Event로 규정하고, 분석을 통하여 Concurrent 기간을 계산하는 것도 가능한 방법임.

# 1 Concurrent Delay 식별

## 5) Time Slice Window Analysis

Window	Projected Completion	Projected Delay (Overall / in Period)	Critical (Longest Path)
August 2013	22 November 2016	0CD/0CD	Near Critical (27CD of Float)
November 2013	22 November 2016	0CD/0CD	Concurrently Critical (1CD of Float)
February 2014	10 April 2017	139CD/139CD	Yes
May 2014	7 April 2017	136CD/-3CD	Yes
August 2014	7 April 2017	136CD/0CD	Yes
November 2014	7 April 2017	136CD/0CD	Yes
February 2015	1 April 2017	130CD/-6CD	Yes
May 2015	22 March 2017	120CD/-10CD	Concurrently Critical
August 2015	22 March 2017	120CD/0CD	Concurrently Critical
November 2015	22 March 2017	120CD/0CD	Concurrently Critical (5CD float)
February 2016	4 March 2017	102CD/-18CD	No mitigation measures taken
May 2016	18 March 2017	116CD/14CD	No (works complete)
August 2016	1 March 2017	99CD/-17CD	No (works complete)
November 2016	1 March 2017	99CD/0CD	No (works complete)
February 2017	7 March 2017	106CD/7CD	No (works complete)

- TSWA는 Fragnet을 넣지 않고 Updated Program의 분석을 통하여 지연기간을 계산함.
- 각 Updated Program의 Critical Path를 구성하는 로직 중 2개 이상의 Activity가 있는 경우 Concurrent Delay로 계산함.
- TSWA는 APAB Window Analysis와 비슷하지만 Cutoff를 기준으로 미래지연을 위주로 결과를 분석하는 방법에 가까움. 즉 CP Activity의 변경을 추적하기 보다는 CP Activity가 결정하는 미래일정을 분석함.
- 그러나 개념적으로 보면 분석방법으로 보기에는 애매하며, 단순하게 Updated Program을 분석하는 수준의 방법임. 따라서 아래와 같이 Updated Program이 정상적으로 유지되지 않는 경우 분석의 신뢰성이 담보되기 어려움.
  - 완료시점이 거의 변하지 않는 경우
  - 로직의 변경이 과도하게 많이 발생한 경우
  - 비합리적인 일정의 조정이 있는 경우

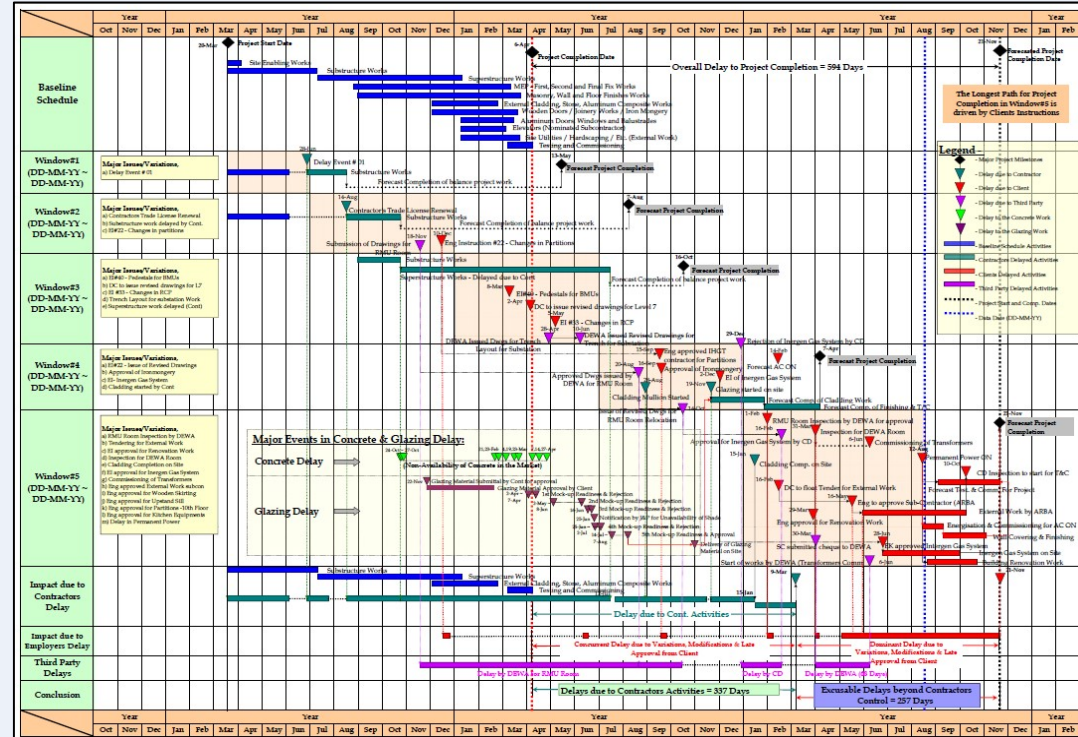
# 1 Concurrent Delay 식별

## 6) As-Planned v As-Built Window Analysis

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

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)

- APAB와 거의 비슷한 방법이나 지연의 기간이 Window Level에서 나눠서 정리됨.
- Window 구간으로 나눠서 지연을 인식한 이후에 그 지연이 발생한 원인을 분석하는 과정에서 Concurrent Delay가 식별될 수 있음.

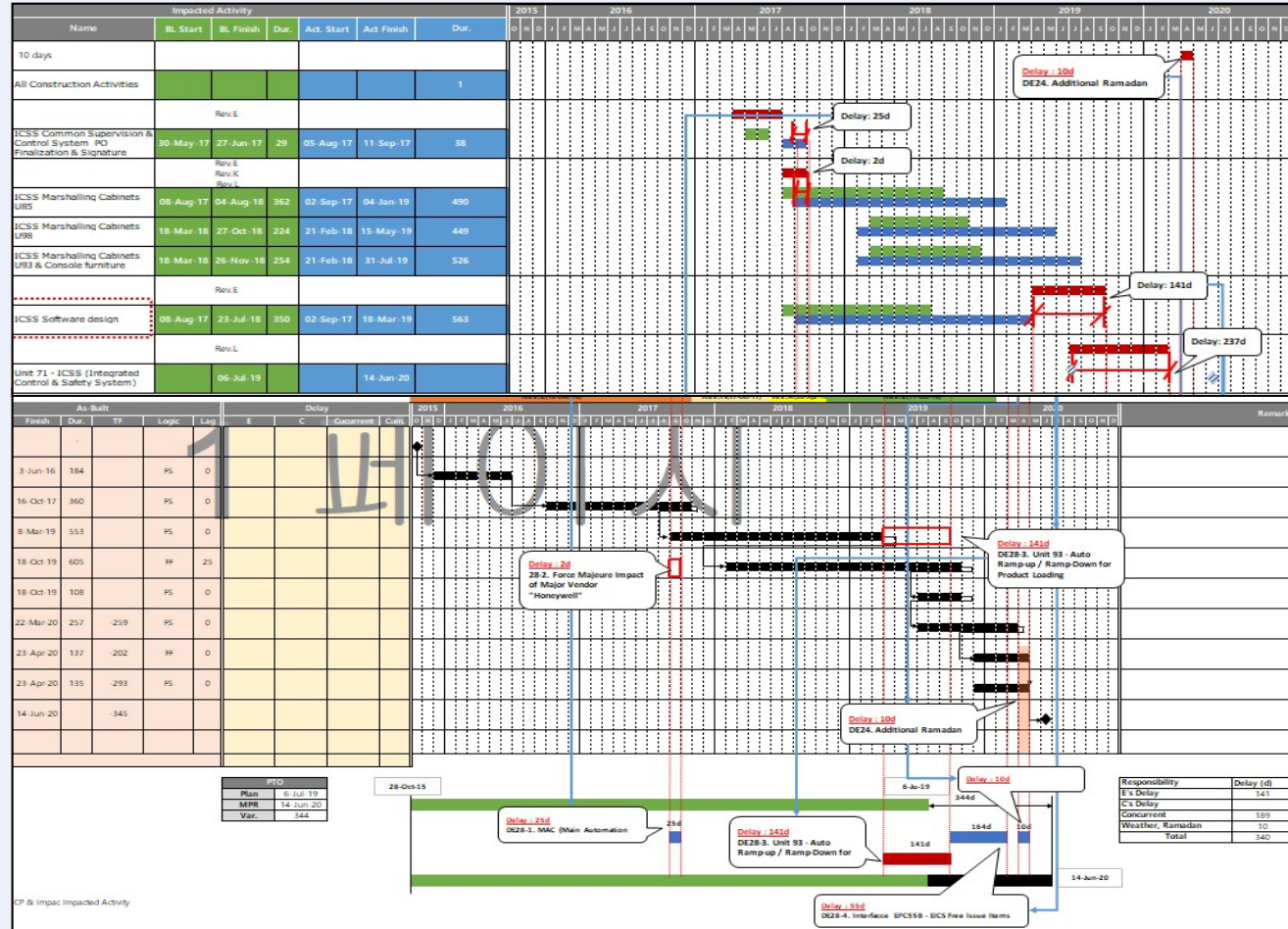


- Updated Program이 없는 경우 사용되기 때문에 세부적인 분석이 쉽지는 않으며, 결국 Baseline의 Float에 따라 Dominant Cause를 결정하고 그 결과에 따라 Concurrent Delay를 판단해야 함.

# 1 Concurrent Delay 식별

## 7) Retrospective Longest Path Analysis

- 먼저 As-Built의 Critical Path를 식별함. 완료시점에서 역으로 CP를 추적하는데, Industry Practice 및 Float를 고려하여 합리적인 경로를 확인함.
- 확인된 As-Built Critical Path 및 Critical Activity의 Plan일정을 같이 비교해서 특정한 구간에 지연이 발생된 것을 인식할 수 있음.
- 그 지연된 구간은 특별한 설명이 있지 않는 한 기본적으로 시공자의 지연임. 따라서 특정한 구간을 지연시킨 발주자의 Event가 있는지 확인 (Event는 Critical Activity와 관련이 있어야 함.) 만약 있다면 그 지연 기간은 발주자의 지연임.
- Critical Activity와 관련이 없는 발주자의 Event가 발생했다면 Concurrent Delay 여부를 확인해야 함.
- 즉 발주자의 Event가 영향을 미친 Impacted Activity의 Float 변화와 As-Built Critical Path상의 Critical Activity의 Float 변화를 비교하여 Concurrent Delay 여부를 확인할 수 있음.



# 1 Concurrent Delay 식별

## 8) Collapsed As-Built Analysis

Event (Fragnet)	Event Type	Event Actual Start Date	Event Actual Finish Date	Event Duration (Days)	CAB Base Schedule	Base Schedule Data Date	Projected Completion Date	Net Loss/Gain	Cumulative delay		
									EDE	CDE	Con-current
001	EDE	7-Sep-08	11-Sep-08	4	CAB	31-Aug-08	18-Jan-09				
					CAB	31-Aug-08	18-Jan-09	0			
002	CDE	4-Sep-08	5-Sep-08	1	CAB	31-Aug-08	15-Jan-09	3		3	
					UD03	31-Aug-08	14-Jan-09	1		1	
003	CDE	15-Aug-08	24-Aug-08	9	CAB	14-Aug-08	9-Jan-09	5		5	
					UD02	31-Jul-08	6-Jan-09	3		3	
004	EDE	2-Jul-08	7-Jul-08	5	CAB	30-Jun-08	1-Jan-09	5	5		
005	CDE	2-Jul-08	5-Jul-08	3	CAB	30-Jun-08	1-Jan-09	0			
					UD01	30-Jun-08	5-Jan-09	-4		-4	
006	EDE	5-Jun-08	8-Jun-08	3	CAB	1-Jun-08	5-Jan-09	0			
007	EDE	31-Jun-08	6-Jun-08	3	CAB	1-Jun-08	2-Jan-09	3	3		
Contract Completion Date:							1-Jan-09	1		1	
<b>Totals</b>								<b>17</b>	<b>8</b>	<b>9</b>	<b>0</b>

- CAB는 Baseline이 필요 없고, As-Built Program을 기본으로 하는 분석방법임. 그러나 As-Built Program이 완료되면 Actual에 맞춰서 로직을 재구성해야 하는 어려움이 있음. 이 점 때문에 CAB를 정확히 사용하기에는 현실적으로 어려움이 있음.
- 그러나 Baseline이 승인 받지 못하거나 없는 상황 또는 Program이 계속 변경되었다면 불가피하게 CAB를 사용할 수 밖에 없는 상황도 있음.
- CAB는 TIA와 비슷한 개념인데, 시간의 흐름만 반대로 진행되기 때문에 Concurrent Delay 식별은 TIA와 비슷하게 이해하면 됨.