SCL Delay & Disruption Protocol – 2nd Edition

• INTRODUCTION

• TIME
  - Methods of Delay Analysis

• OTHER MATTERS
  - Concurrent Delay
  - Records & Programmes
  - Disruption
Introduction

• Background to SCL Protocol
• Reasons for 2nd Edition
  - Developments in the law and construction industry practices
  - Feedback on the Protocol since 2002
  - Developments in technology since 2002
  - The scale of large projects having increased, leading to a wider divergence between small scale and large scale projects
  - Use for international projects as well as UK projects.
Time
Time Claims

• The biggest obstacle to successful time claims is the quality of the programme.

• Records! Records! Records!

• Examples:
  - Milestones not connected to any driving activities
  - Missing actual progress
  - No connection between what the programme shows and what is claimed
Good Programming Practice

- A programme should show:
  - Contractual start and finish dates
  - Access dates
  - Key dates / milestones
  - Planned completion milestones
  - Key resources
  - Key relationships
  - External works e.g. Govt, utilities
  - Critical activities / path
Good Programming Practice (Cont’d)

• Use simple logic
  • Use another activity instead of a logic lag

• Don’t litter the programme with constraints
  • They restrict knock on effects from being visible

• Do record:
  • Actual starts
  • Actual finishes
  • Remaining duration

• Do re-analyse after each programme update
Delay Analysis

• Attempts to ascertain what has happened and establish the facts.

• Attempts to find out what the effect of the claimed delay is, or the cause of the identified effect.

• Attempts to demonstrate for either side of an argument, that theirs is the correct interpretation.

• Tries to form a conclusion and answer.

• Informs the Court / Tribunal
## Six Methods of Delay Analysis in the Protocol

<table>
<thead>
<tr>
<th>Method of Analysis</th>
<th>Analysis Type</th>
<th>Critical Path determined</th>
<th>Delay Impact determined</th>
<th>Requires:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impacted As-Planned Analysis</td>
<td>Cause &amp; Effect</td>
<td>Prospectively</td>
<td>Prospectively</td>
<td>Logic linked baseline programme. A selection of delay events to be modelled.</td>
</tr>
<tr>
<td>Time Impact Analysis</td>
<td>Cause &amp; Effect</td>
<td>Contemporaneously</td>
<td>Prospectively</td>
<td>Logic linked baseline programme. Update programmes or progress information with which to update the baseline programme. A selection of delay events to be modelled.</td>
</tr>
<tr>
<td>Time Slice Windows Analysis</td>
<td>Effect &amp; Cause</td>
<td>Contemporaneously</td>
<td>Retrospectively</td>
<td>Logic linked baseline programme. Update programmes or progress information with which to update the baseline programme.</td>
</tr>
<tr>
<td>As-Planned versus As-Built Windows Analysis</td>
<td>Effect &amp; Cause</td>
<td>Contemporaneously</td>
<td>Retrospectively</td>
<td>Baseline programme. As-built data.</td>
</tr>
<tr>
<td>Longest Path Analysis</td>
<td>Effect &amp; Cause</td>
<td>Retrospectively</td>
<td>Retrospectively</td>
<td>Baseline Programme. As-built programme.</td>
</tr>
<tr>
<td>Collapsed As-Built Analysis</td>
<td>Cause &amp; Effect</td>
<td>Retrospectively</td>
<td>Retrospectively</td>
<td>Logic linked as-built programme. A selection of delay events to be modelled.</td>
</tr>
</tbody>
</table>

Source: SCL Protocol V2

The Contracts Group Ltd.
1. Impacted As Planned Method

- Cause and Effect analysis
- Takes the original plan, applies the claimed delay event and attempts to show the impact.

Original Plan
Finish Day 15

Variation

EOT = 10D
New Completion = D 25
NO LDs

Original Plan
Finish Day 25
2. Time Impact Method

- Cause and Effect analysis
- Takes the progressed plan, applies the claimed delay event and attempts to show the impact.

![Diagram showing Original Plan Finish Day 15, Variation, As Built Plan Finish Day 23, with EOT = 10D, New Completion D25, NO LDs]
2. Time Impact Method (Cont’d)

- EOT = 30 - 15 = 15D
- EOT Awarded = 10D  New Completion Date = D 25
- LD’s = 30 - 25 = 5D

As Built Plan
Finish Day 30
3. Collapsed As Built (‘but for’) Method

• Cause and Effect analysis

• Takes the as-built programme and subtracts delay events to show what would have been but for the delays.

• Requires a full linked as built programme.

• Only measures incremental delays to the critical path.
4. Time Sliced Window Method

- **Effect and Cause analysis**

- Examines each monthly updated programme to assess the delay for that month to the critical path

- Looks for the causes of the delay.
5. As Planned vs As Built Window Method

- Effect and Cause analysis

- Breaks the original programme into time windows.

- Compares the original critical path in each time window with the as built records to determine what might have caused the delays.

- Is not reliant on programming software.
6. Longest Path Method

- Effect and Cause analysis

- Takes the as-built critical path and attempts to back analyse and determine causes of delay.

- Compares key dates along the longest path to see the deviation from the baseline.

- May not identify or account for changing critical paths.

- Can miss contractual EOT events with no critical impact e.g. adverse weather
## Comparison of Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Simple</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impacted As Planned</td>
<td>✓</td>
<td>• Based on single programme</td>
<td>• Theoretical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Rarely keep to plan</td>
<td>• Rarely keep to plan</td>
</tr>
<tr>
<td>Time Impact</td>
<td>x</td>
<td>• Needs multiple updates</td>
<td>• Time consuming</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Suitable for Linear projects</td>
<td>• Costly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Identifies shifts in critical path</td>
<td></td>
</tr>
<tr>
<td>Time Slice Windows</td>
<td>x</td>
<td>• Needs multiple updates</td>
<td>• Time consuming</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Suitable for complex projects</td>
<td>• Costly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Identifies shifts in critical path</td>
<td></td>
</tr>
<tr>
<td>As Planned vs As Built Windows</td>
<td>x</td>
<td>• Needs multiple updates</td>
<td>• Time consuming</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Suitable for complex projects</td>
<td>• Costly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Identifies shifts in critical path</td>
<td></td>
</tr>
<tr>
<td>Longest Path</td>
<td>(√)</td>
<td>• Requires as built programme</td>
<td>• Cannot determine criticality at times of events</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Suitable for linear projects</td>
<td></td>
</tr>
<tr>
<td>Collapsed As Built (‘As built but for’)</td>
<td>✓</td>
<td>• Based on single programme</td>
<td>• Needs good accurate records</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Based on actual build times</td>
<td>• Cannot determine criticality at time of event</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Does not require baseline &amp; updates</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Suitable for linear project</td>
<td></td>
</tr>
</tbody>
</table>
Other Matters
Other Matters

• In addition to programming and delay methodologies SCL review for 2\textsuperscript{nd} Edition limited to:
  - Records
  - Global claims and concurrent delay – in light of recent English case law
  - Approach to consideration of claims (prolongation / disruption – time and money) during currency of project
  - Model clauses
  - Disruption

• Model Clauses have been dropped from the 2\textsuperscript{nd} Edition
Concurrent Delay

- Protocol attempts to define “Concurrent delay” and provides a recommended view
- Has the SCL missed an opportunity to simplify this matter?
Concurrent Delay (cont’d)

Protocol Example [3.10.7]

Contractor event causes 5 week delay:

5 week delay to completion.

Variation (Employer event) causes 2 week delay:

In absence of Contractor delay would have caused 2 week delay to completion.

Question – Should the Contractor be awarded 2 weeks EOT?
Concurrent Delay (Cont’d)

• The Protocol acknowledges that it can be argued either way, but recommends that the variation should be seen as not causing delay to completion (and therefore there is no concurrency and no EOT)

• This principle is questioned by some lawyers:
  - “definition of concurrency may be unnecessarily restrictive”
  - “favours whichever delay comes first”
  - “takes no account of prevention principle”
Concurrent Delay (Cont’d)

• The alternative view: Both delays, in the absence of the other delay, would be effective causes of delay. They are concurrent. EOT should be awarded thereby avoiding the ‘windfall’ which otherwise accrues to the Employer.

• In my experience Engineers in Hong Kong will generally award a 2 week EOT.
Records & Programmes

• Guidelines on maintenance and storage of records including programmes
• Recommendations include agreement of records up front, and consideration of:
  (a) The types of records to be produced and the information to be contained therein
  (b) Who is responsible for both producing and checking those records
  (c) The frequency with which those records are to be updated or produced
(d) The format of those records (for example, to ensure compatibility with any project-wide database)

(e) The ownership (including any relevant intellectual property rights) and storage of, and access to, those records

• The 2nd Edition takes 6 pages to list and describe records, compares to 2 paragraphs in the 1st edition. Also Appendix B includes 8 pages, lists and describes typical records which are recommended.
Records & Programmes (Cont’d)

• The Protocol also deals with categories of records, including:
  - Programme
  - Progress
  - Resources
  - Costs
  - Correspondence & Administration
    - Letters/emails
    - Contract Management
    - Technical
    - Milestones
    - Claims
  - Contract & tender documents
Records & Programmes (Cont’d)

• Programme
  - Form of the Contractor’s proposed programme
  - Detail within the proposed programme
  - Interaction with method statements
  - Time for submission
  - Mechanism for obtaining acceptance of proposed programme
  - Updating and saving programmes

• The Protocol only recognises the growing use of BIM; no specific recommendation regarding use.
Disruption

• Disruption can be difficult to prove, and to be paid. Further SCL guidance is welcome.

• Much expanded in 2nd Edition with own section. Now includes ‘Methods of disruption analysis’.

• 9 methods noted – 7 productivity based methods and 2 cost based methods. The Protocol describes each method.

• ‘Measured mile’ analysis is still the preferred method.
  - Compares productivity of disrupted part of the works with similar works not disrupted.
Disruption (Cont’d)

• Other methods, in descending order of preference:

  Project Specific Studies
  (a) Earned value analysis
  (b) Programme analysis
  (c) Work or trade sampling
  (d) System dynamics modelling

  Project Comparison Studies

  Industry Studies
Disruption (Cont’d)

Cost – based methods

(a) Estimated v incurred cost

(b) Estimated v used labour
THE CONTRACTS GROUP LTD.

Thank You