



Part 1 – Float mapping and the extensions of time

It is well established that in most contracts the planned critical path will change many times during the life of the project. One of the problems when using a planned v as-built method of delay analysis, particularly on a pure observational basis, is that it is not often clear where the critical path has been or when it changed. It is equally tricky to identify if there are concurrent delay paths that might affect the assessment of compensation. Float mapping helps with this problem by setting out, in easily understood format, where these critical and near critical paths lie in the as-built programme.

When computer programming software is used to manage a project it forecasts a completion date for the work through the longest path of activities. Where this path has no float it is said to be a critical path. At each reporting date (data date) actual start dates, progress and actual completion dates are recorded and the work is re-scheduled so that work completed (as-built) is on the left side of the data date and remaining work (forecast) on the right. The computer can only calculate the critical path through the forecast network of remaining activities. On the left side actual dates are recorded and logic links are effectively removed. At PC, what is left is the as-built programme with all completed work on the left side of the data date with no computable critical path. Even if the final as-built programme could be redrawn with logic it might not reflect the changing position of the critical path with time, it would only reflect the final critical path.

However, if progress reporting has been carried out regularly there will be an auditable trace of progress for each activity at each data date. A product of this progress, assuming there is a fully logic linked programme, is also an auditable trace of float for each activity at each data date. This float will most likely change in each report and determine whether or not an activity lies on the critical path or near to it. By simply mapping these float values on a spreadsheet we can see how the float changes and where the critical activities with the lowest float values are lying.

Contact:

To discuss in further detail the issues raised please contact:-

Bob Cooper

Director of Disputes
 bob.cooper@systech-int.com
 +44 (0) 20 7940 7656



There are basically two ways that the programme can be set up to display float. If the Contract Completion Date is fixed (constrained) then any work paths that extend beyond this date will show as negative float. If the Completion Date is not constrained then the critical path will always show as zero float and will drive out the Planned Completion Date according to the longest/critical path.

This is a matter of preference and for this short blog we will assume the former because negative float is easier to read from the juxtaposition of planned and as-built bars in the example below.

Let's assume that the planned v as-built programme looks something like this;

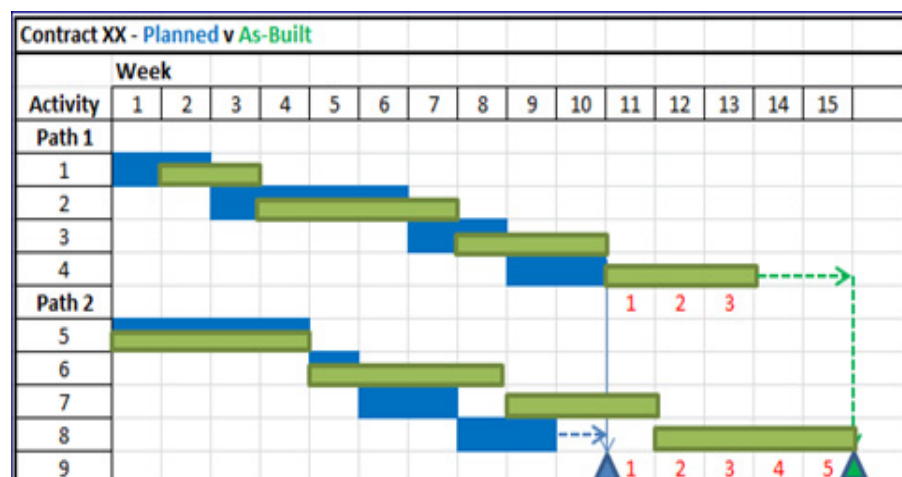


Figure 1: Planned v As-Built Programme with a final delay of 5 weeks.



Part 1 – Float mapping and the extensions of time

From a simplistic observational view we might assume that because path 2 is driving the practical completion date it was solely responsible for the overall delay. However, by using a float map the true position is revealed as significantly different. Here are the float values from each progress report tabulated in a spreadsheet. The areas shaded in green are the as-built periods for each activity. When an activity is completed no float values appear thereafter because the activity will appear to the left of the data date with logic removed so no float will be available from the computer programme;

The positive values indicate that the activities have float (not critical), the zero values indicate that all float has been absorbed (critical but not yet causing delay) and the negative values indicate delay in relation to the Contract Completion Date. As the Contract Completion Date has been constrained, the activity with the largest negative float value will be the driving (critical) activity. The next step is to identify in which week and on what activity the delay occurred and who owns the risk under the Contract. The float map helps us to carry that out. In Figure 2 the weeks when the delay (negative float) on each activity occurred are shaded red and the liability indicated in the cell above; E = Employer and C=Contractor.

It is a general principle in many jurisdictions that concurrency should not alter the Contractor's entitlement to time. In Figure 3 Employer delays E1 and E2 both have a concurrent effect of 1 week (float -1). E3 and C1 both have concurrent effect causing a further 1 week delay (float -2). C2 and C3 similarly have concurrent effect causing a further 1 week delay (float -3). E4 and C4 are sequential delays (float -4 and -5). The Contractor's EoT award in this instance would be 3 weeks (E1/E2 +E3+E4) irrespective of the Contractor's concurrent delays C1 and C2/C3.

Activity	Week														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Path 1															
1	-1	-1	-1	< activity complete											
2	-1	-1	-1	-1	-1	-1	-1								
3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2					
4	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	-2	-2	-3		
Path 2															
5	1	1	1	1											
6	1	1	1	1	1	0	-1	-2							
7	1	1	1	1	1	0	-1	-2	-2	-2	-3				
8	1	1	1	1	1	0	-1	-2	-2	-2	-3	-3	-3	-4	-5
9															

Figure 2: Float Map - Values

Activity	Week														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Path 1															
1	-1	-1	-1												
2	-1	-1	-1	-1	-1	-1	-1			C1					
3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2			C3		
4	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	-2	-2	-3		
Path 2															
5	1	1	1	1			E2	E3							
6	1	1	1	1	1	0	-1	-2		C2					
7	1	1	1	1	1	0	-1	-2	-2	-2	-3			E4	C4
8	1	1	1	1	1	0	-1	-2	-2	-2	-3	-3	-3	-4	-5
9															

Figure 3: Mixed delay events on both Paths

Conclusion

Analysis of delay in this way still relies on a sound CPM Programme with regular progress updates in order to provide reliable float values. In the real world, where programmes may have thousands of activities, the analysis will inevitably be far more complex than the simple example above. In part 2 we will consider the use of float mapping for assessment of compensation.