gbaprojects
minimising risk, enhancing performance

THE POOR PLANNING PRACTICE OF USING NEGATIVE LAG:

BETTER PLANNING METHODS IN MICROSOFT PROJECT

Matt Betros, B.CIS, B.MNGT, Dip.PM, MAIM
Director GBA Projects Pty. Ltd.

INTRODUCTION

Here at GBA Projects we have a general rule of thumb, do not use negative lag. Why? To put it simply, using negative lag breaks the whole idea of using the logic type in the first place. Logic is specifically designed to allow a forward and backward pass calculation in order to determine early and late dates followed by subsequent free and total float. This means that the logic allows us to define the successors (for the forward pass) and predecessors (for the backward pass).

Without logic, a programme is merely a collection of lines on a chart without much meaning. A schedule should clearly identify a project's total scope of work and the relationship between each package of work in a manner that is transparent.

To achieve this, we must use the appropriate logic and activities to model our execution plan on paper and subsequently allow for control of the execution of works. Many planners argue that this takes time and this leads to the use of negative lag to cut down both the complexity of a schedule and the number of activities within it.

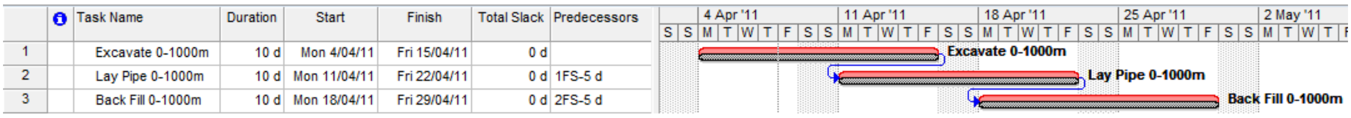
A common argument which favours the use of negative lag is that the alternatives available lead to schedule 'bloating'. Many schedulers argue that if they were to have 100 tasks in their network for any given project that use a negative lag relationship, they would need to double the size of their project to accommodate proper planning logic. This is an excuse that highlights poor planning practice in today's projects.

There are other alternatives available that whilst increasing the size of a project schedule also add an element of transparency (understanding the programme) and control. Calvey (2006) suggest that fewer "…activities, by definition, create[s] activities with more work scope taking place within those activities and that masks the project's work plan and additionally can complicate the ability to accurately status the work.". This promotes the ideal that a schedule is designed to capture the whole scope of work and this includes all logic which drives our activities.

This paper aims to address the negative lag concept and offers viable solutions to planners in order to provide a more truthful, transparent and accurate schedule. This leads to a greater understanding of float and a closer relationship to project time control.

OUR EXAMPLE

In order to demonstrate this issue, we are going to use Microsoft Project and a simple 3 activity network. Suppose our demonstration included the following scenario:



	ℹ	Task Name	Duration	Start	Finish	Total Slack	Predecessors
1		Excavate 0-1000m	10 d	Mon 4/04/11	Fri 15/04/11	0 d	
2		Lay Pipe 0-1000m	10 d	Mon 11/04/11	Fri 22/04/11	0 d	1FS-5 d
3		Back Fill 0-1000m	10 d	Mon 18/04/11	Fri 29/04/11	0 d	2FS-5 d

Figure 1: showing a typical planning scenario using a negative lag

Examining the logic above, the planner has attempted to indicate 'lead up time' from each activity to it's subsequent activities. Assume in the example we had worked out that the required lead time was 5d for each task. The planner would typically introduce a negative lag of 5 days to bring each successor forward in time. Also note too that our total float (slack) is defined as zero.

If we think about this relationship, we have stated to Microsoft Project that *Lay Pipe* cannot *start* until *Excavate* is *complete*, yet in using the negative lag we've forced the start of pipework to occur *before* the completion of excavation thus breaking the rule governing the relationship.

If we were to examine this closely and translate it onto a time chainage chart, we would note the following linear balance chart:

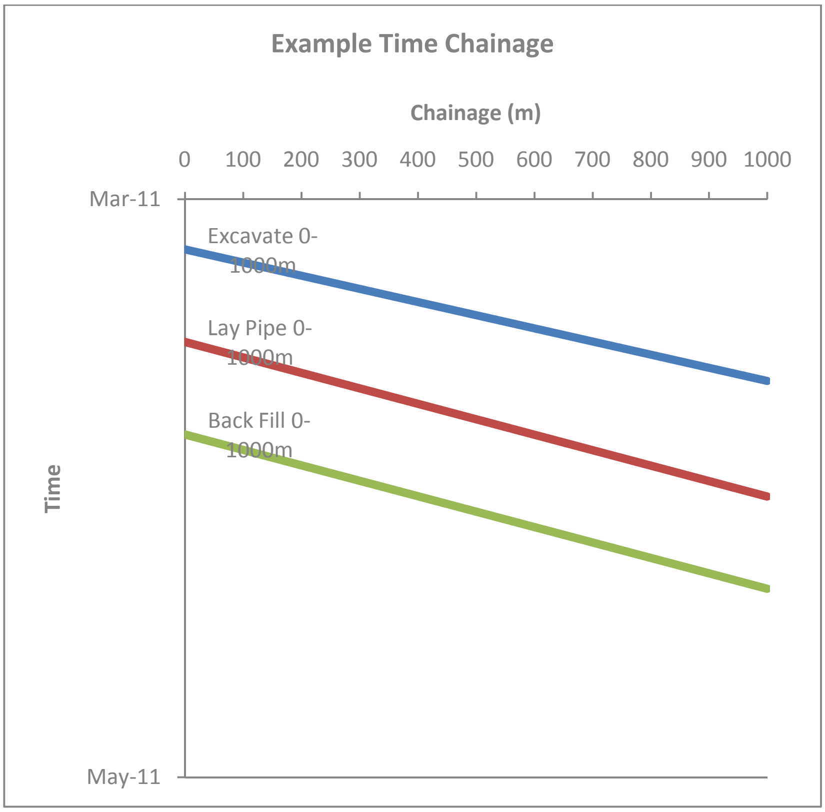
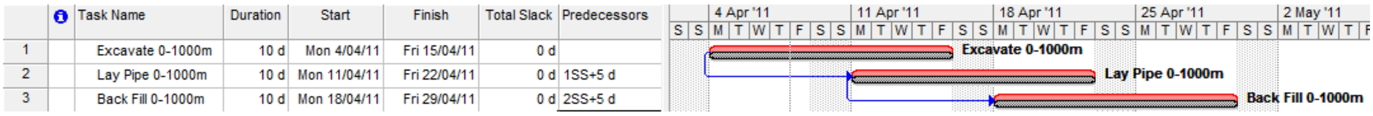


Figure 2: showing a linear balance chart of work rates

Now one might argue that in reality, the excavator would progress a lot quicker and this is true, but for the sake of the paper, we are assuming a more linear and equal rate of work. The balance chart allows us to note the logic that is required here. Clearly, it promotes the original intent of the logic.

Before discussing options to fix our above example, we must introduce another rule that we have in our planning guidelines which is that any task having a Start-to-Start successor should also have a Finish-to-Finish successor to that same task.

If a task is linked with a Start-to-Start relationship, it generally indicates that there exists some form of finishing relationship. Continuing our scenario we replace our previous logic with something more reflective of the true progression of works:

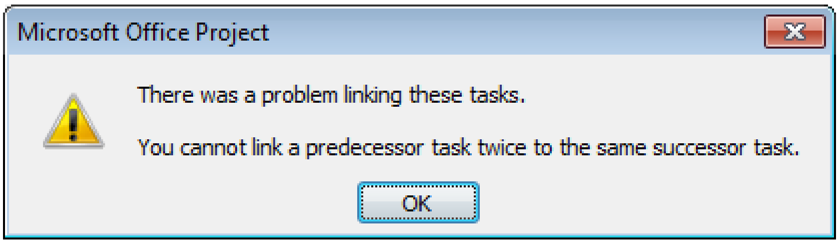


	Task Name	Duration	Start	Finish	Total Slack	Predecessors
1	Excavate 0-1000m	10 d	Mon 4/04/11	Fri 15/04/11	0 d	
2	Lay Pipe 0-1000m	10 d	Mon 11/04/11	Fri 22/04/11	0 d	1SS+5 d
3	Back Fill 0-1000m	10 d	Mon 18/04/11	Fri 29/04/11	0 d	2SS+5 d

Figure 3: showing a typical planning scenario where a SS link is used in place of negative lag

In our example, we commence our excavation work for our trench and then 5 days later we start laying our pipes. In this manner, we *cannot start laying pipes until we have started to excavate*. Once we have *finished* excavating we will be able to *finish* laying all of our pipes because our excavation plant & equipment will have been removed. This therefore indicates that we require a Finish-to-Finish relationship with a 5 day lag between the two tasks.

However, in trying to achieve this, Microsoft Project will throw the following error:



Microsoft Office Project

There was a problem linking these tasks.

You cannot link a predecessor task twice to the same successor task.

Figure 4: showing the error users receive when allocating the same predecessor multiple times

The product is preventing us from making a second predecessor link from the same task. This is a strange functionality for a planning package considering that applying the aforementioned logic is a planning and scheduling standard!

Microsoft Project does not allow the appropriate linkages to be included between the activities allowing for oversight on the part of the scheduler and misreporting by the software itself. The better way to treat the problem is to prevent it’s cause. The planning package should be able to accept the delay and adjust the forecast appropriately to avoid potential mis-reporting.

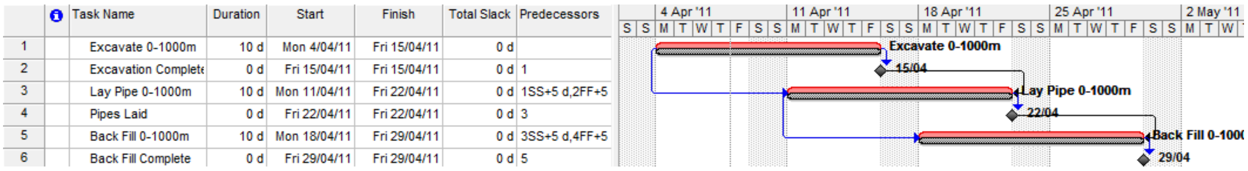
ALTERNATIVE OPTIONS

Paul Harris (http://www.planningplanet.com/forums/primavera-project-planner-p3/414072/negative-lag), a noted commentator in planning software, suggests that the quantum of lag to be included within the relationship is best placed somewhere within the activity. This seems more appropriate given that a relationship is supposed to describe the logic between two activities and not necessarily a time period between them. It is a representation of sequence not duration specifically.

So how could we represent this network in a more logical manner? There exists three options, all of which we discuss in this paper.

Milestones

To introduce our first option, we include a milestone between each activity to break up the programme. Each task has a milestone signifying the completion of it and this allows us to define a Finish-to-Finish relationship to a related task.



	Task Name	Duration	Start	Finish	Total Slack	Predecessors
1	Excavate 0-1000m	10 d	Mon 4/04/11	Fri 15/04/11	0 d	
2	Excavation Complete	0 d	Fri 15/04/11	Fri 15/04/11	0 d	1
3	Lay Pipe 0-1000m	10 d	Mon 11/04/11	Fri 22/04/11	0 d	1SS+5 d,2FF+5
4	Pipes Laid	0 d	Fri 22/04/11	Fri 22/04/11	0 d	3
5	Back Fill 0-1000m	10 d	Mon 18/04/11	Fri 29/04/11	0 d	3SS+5 d,4FF+5
6	Back Fill Complete	0 d	Fri 29/04/11	Fri 29/04/11	0 d	5

Figure 5: showing the milestone method to allow a more appropriate network

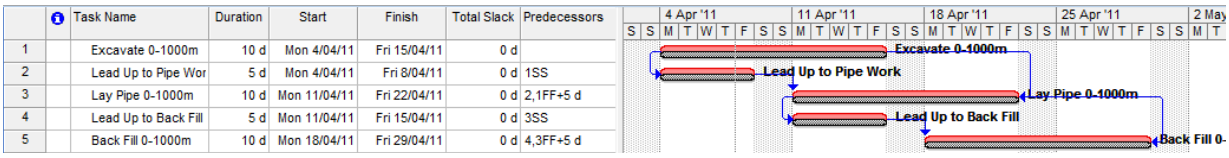
You will quickly notice that we've doubled the size of the programme but still managed to obtain the same float value. Unfortunately there is no way around this so called 'schedule bloating' because of our software restrictions.

This milestone method is usually preferred because it is quick to setup and also allows for additional reporting to be provided to the client due to the additional milestones that can be reported against.

Lead Up tasks

If we consider our example above, the relationship of Start-to-Start between the Excavation activity and the Lay Pipe activity is quite simple. We must wait 5 days (which we have already calculated) before we can start laying the pipe work otherwise we run the risk of clashing between our excavation crew and our pipe laying crew.

If we consider the following amendment to the schedule, we are able to model the relationships a lot better:



	Task Name	Duration	Start	Finish	Total Slack	Predecessors
1	Excavate 0-1000m	10 d	Mon 4/04/11	Fri 15/04/11	0 d	
2	Lead Up to Pipe Wor	5 d	Mon 4/04/11	Fri 8/04/11	0 d	1SS
3	Lay Pipe 0-1000m	10 d	Mon 11/04/11	Fri 22/04/11	0 d	2,1FF+5 d
4	Lead Up to Back Fill	5 d	Mon 11/04/11	Fri 15/04/11	0 d	3SS
5	Back Fill 0-1000m	10 d	Mon 18/04/11	Fri 29/04/11	0 d	4,3FF+5 d

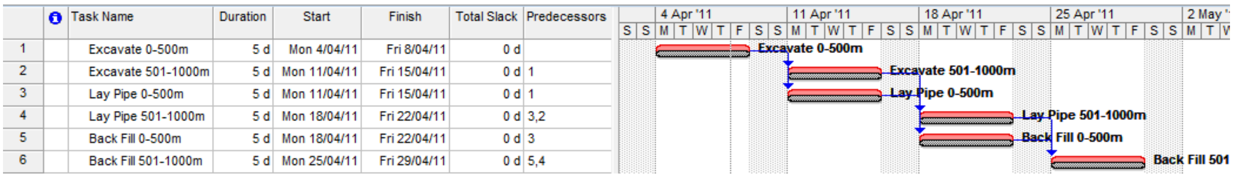
Figure 6: showing the lead up time as a task rather than a lag on a relationship

Now again note that our float values have not changed and we are able to readily see the intent of the original logic. Not only that, but we have managed to minimise the amount of relationship lag which makes for a more transparent and readable schedule. It also allows for a greater control because we can monitor the Excavation works (for example) and understand whether the Lead up Time to Pipe Laying will increase or reduce.

Task Detail Definition

Possibly the better option of the three available is to actually break the activities into relevant detail to properly map the logic between the types of work to be performed.

We already worked out our rate of work (100m per day) and if we therefore consider this, it equates to 500m in 5 days (our lead up time). So consider splitting the activities as follows:



		Task Name	Duration	Start	Finish	Total Slack	Predecessors
1		Excavate 0-500m	5 d	Mon 4/04/11	Fri 8/04/11	0 d	
2		Excavate 501-1000m	5 d	Mon 11/04/11	Fri 15/04/11	0 d	1
3		Lay Pipe 0-500m	5 d	Mon 11/04/11	Fri 15/04/11	0 d	1
4		Lay Pipe 501-1000m	5 d	Mon 18/04/11	Fri 22/04/11	0 d	3,2
5		Back Fill 0-500m	5 d	Mon 18/04/11	Fri 22/04/11	0 d	3
6		Back Fill 501-1000m	5 d	Mon 25/04/11	Fri 29/04/11	0 d	5,4

Figure 7: splitting our activities into a little more detail

What we now see is a complete change in the way the programme is presented. We can clearly see that by representing the first 500m of work by three tasks we are now able to control the programme with greater accuracy but also allow for greater transparency. A planner picking this programme up for the first time will quickly understand that we intend to start laying pipes after 500m of excavation and so forth.

Synopsis

In summary, there exists a common trend with Microsoft Project users which is quite common across the world. These schedulers simply ignore the requirement for proper planning logic and insert negative lags to achieve a quick programme with minimal activities.

There are other alternatives as we have discussed which add more tasks to a schedule, but as these individuals would contest, creates a “bloating” effect in the programme. This is not an excuse for good planning practices. Planners should be vigilant to prepare a programme that makes sense and is logical. However as we’ve noted, logic itself is merely a representation of how we intend to execute work not necessarily how long.

Minimising the use of lag increases transparency within a schedule which is truly a good thing to apply to any programme. By breaking a programme up into more detail and replacing lags with tasks we are able to understand the programme better and increase the control we have over the execution of the works.

But how does each option work under progress?

APPLYING PROGRESS

We've already noted that we can achieve the same sequence of works using better planning methods, but how does each method compare when progress is applied to them? This section of our paper aims to find out.

Setup

For our example, we will assume that week one of excavation went well and we were on target, managing to complete 500m of excavation. However in week 2, we encountered a 2 day delay as the excavator broke down on the 500-1000m run. With the status date at 15-Apr, lets examine how progress affects each planning option.

Keep in mind our schedule completion date is 29th April.

Negative Lag

We assume that we are on target through week 1 and complete our first 500m, however into week 2 we experience our delay of 2 days. Suppose we manage to start laying the pipe on the Monday as planned but experience a delay on the final stretch.

We must update our programme to reflect the current status:

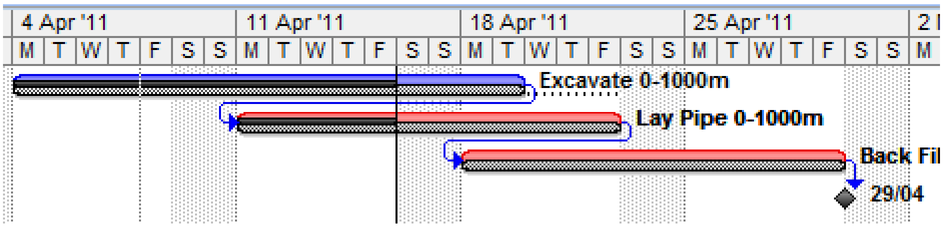


Figure 8: we model the current progress and add a two day delay. Note the end date has not moved

After inserting progress for the excavation and pipe work, we can clearly note that the delay is not modelled correctly the end date still shows 29-Apr and not 3-May as it should (with the delay event). In addition, the excavation task was critical and now shows that it is not critical. The 2 days of float we lost in the delay has actually been inserted as float into the activity which was delayed.

Milestones Approach

If we examine the same progress update against our milestones option, we can note that the delay is modelled correctly. We insert our actual start dates for both excavation and pipe laying along with the actual and original durations.

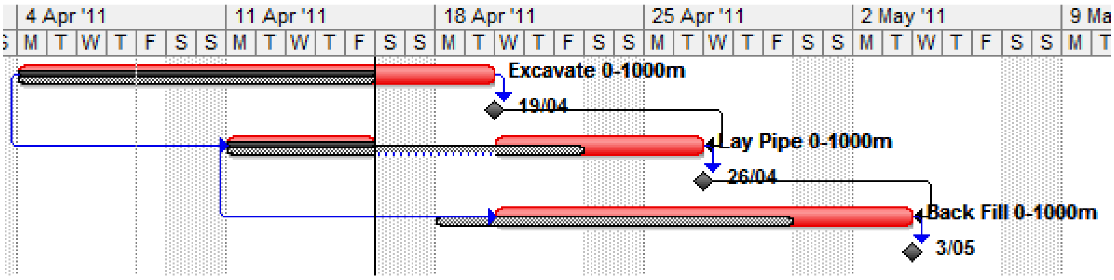


Figure 9: we now see the finish-to-finish logic playing a vital role in forecasting

We can now see that the delay of 2 days is modelled such that the end date for our mini project is extended by two days, a true representation of what has occurred.

We should however, be able to start back filling the first 500m because that work has been complete. We can note in our example that the commencement of backfilling has shifted by 2 days due to our delay event and the Finish-to-Finish logic applied to the activity itself. This in itself is not a major problem because when we come to apply progress for the consecutive week, we will see actual dates inserted but the forecast finish will remain the same and include the delayed duration.

Lead Up Time Approach

Using our lead up time option we are able to model the same outcome with a greater deal of control.

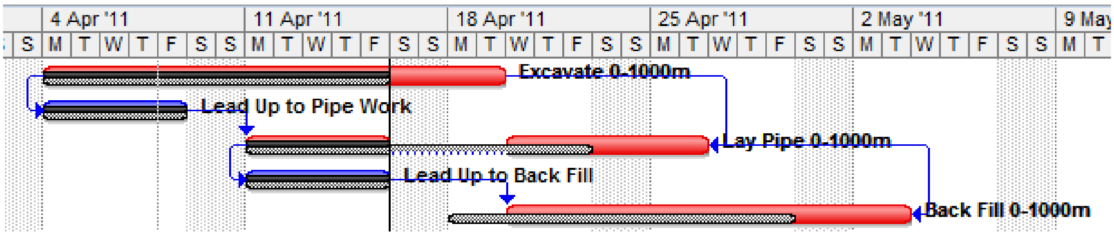


Figure 10: updating our more detailed model still poses a few control issues

However there is one minor issue with our programme updated as it is. Similar to the milestones approach, we also note that the lead up time for back filling is also complete which means we are allowed to start back filling the first 500m. In reality this is true because the first 500m have not been affected by the delay. If we then examine the schedule, we note that the day slippage has been modelled, however the scheduled dates for the following week are not quite correct. We should be starting the Back Fill on the Monday.

However if we were to model the progress for week three on this level of detail, we would note the actual start and would still receive the same delay forecast as we have incurred it.

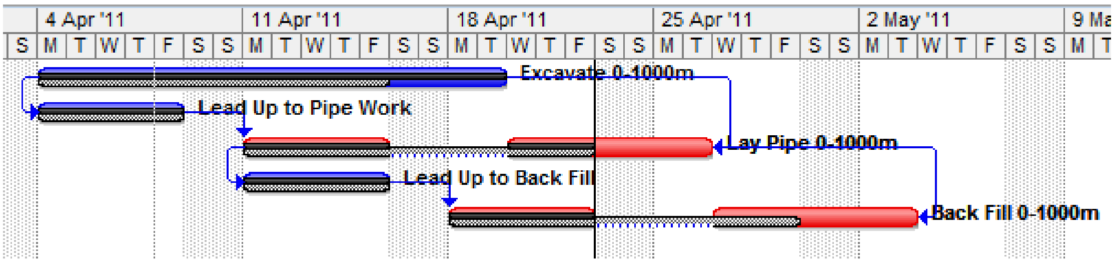


Figure 11: we update the progress as planned in week 3 and not the slippage now affecting the backfill

After applying week 3 progress to the network, we noted that our actual start for backfill was indeed Monday 18th however the delay previously incurred in the excavation of the remaining 500m of trench is now noted in that task itself during week 4.

Applying this procedure to the milestones and lead up time alternatives, we find the same result.

Detail Level Progress

But what of our more detailed programme? How does this fare in the grand scheme of alternatives? By applying the same progress information into the programme we actually are left with a variation once more to our progress.

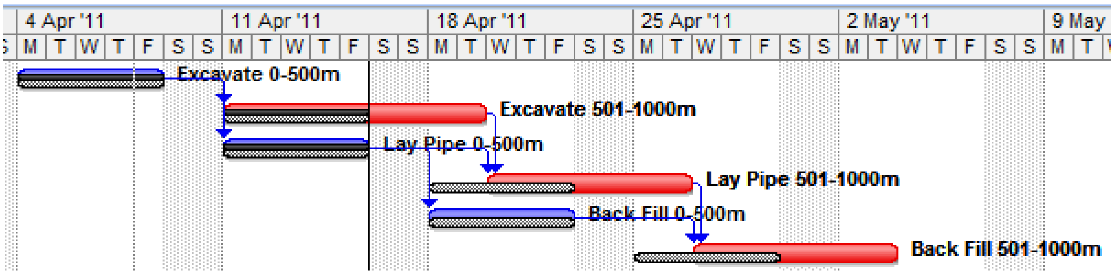


Figure 12: noting delay to a certain area and it's flow on effect

If we examine the progress to this alternative scheduling method, we can note completion of excavation works up to 500m which means by the end of week 2 pipe work for the first 500m is also complete which is a level of detail not seen in the other methods.

In addition, the 2 day delay in the final 500m of excavation clearly shows a trend as the logic pushes the final 500m of pipe work and back filling out into the 5th week. Our duration has increased by 2 days thus proving our case for delay.

More importantly is the fact that the activity for Backfill of the first 500m is correctly forecast where as previous models delayed it's commencement by 2 days. The result is that we gain a full understanding of progress to date, a re-forecast of work to be done in the future with a great deal of accuracy and we also are able to simulate the full effect of the delay event.

If we were to then progress through week 3 as we did with our lead time example, we would notice even more detail about the progression of work.

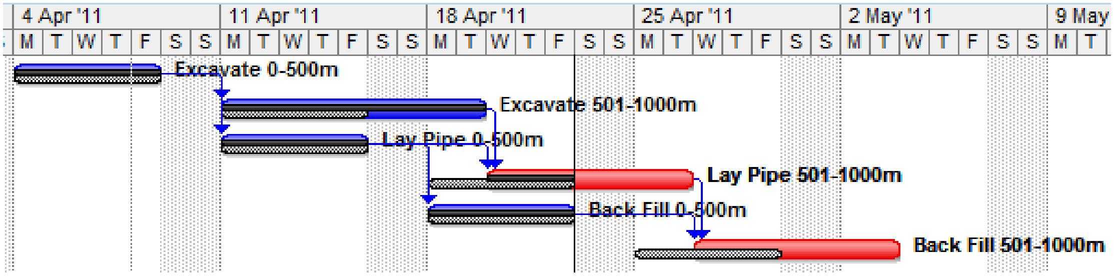


Figure 13: week 3 shows progress and flow on effects for last 500m of work

We can clearly note that the backfilling activities took place on time and we commenced pipe work two days late due to our delay event.

This is clearly the better option.

CONCLUSION

Clearly the best method of including a Start-to-Start/Finish-to-Finish relationship in Microsoft Project is to make use of the detail within the scope of work. For work packages of this nature where one scope of work is reliant on the start and finish of another, it is better planning practice to break out the tasks into more detail.

The benefits of this approach include:

- First and foremost the intent of the logic is upheld and reduces confusion over the sequencing of work
- Lags are replaced by real time activities which adds another level of transparency to the overall project schedule
- Progress can be applied and we can virtually guarantee an accurate forecast will be maintained

Whilst more activities are added into a schedule and a programme can potentially 'bloat', it is certainly no excuse for poor planning. The Microsoft product is but a means to an end and if garbage information is put into the schedule then garbage information will result as output

REFERENCES

Douglas, E, Calvey, T, Donald F, McDonald D, Winter, R, 2006, The Great Negative Lag Debate, http://www.calveyconsulting.com/files/PS02_Great_Negative_Lag_Debate.pdf

Weaver, P, Float is it Real?, 2006, http://www.mosaicprojects.com.au/PDF_Papers/P043_Float%20-%20Is%20it%20Real.pdf