Full Monte Conditional Branching Example

Prerequisites to use the example

Microsoft Project 2010 or later
Barbecana's Full Monte 3.3 or later

Introduction

Conditional Branching allows different tasks to be executed in the schedule based on the date a specific task finishes during the simulations.

The predominant use for this technique is to model risk mitigation strategies.

Pre-Mitigation Example

File: Full Monte Conditional Branching Pre-Mitigation.mpp

The Pre-Mitigation example schedule is shown in Figure 1.

Task Name	Duration	Start	Finish	Risk Assessment	Finish Percentile (100%)	Chance of Achieving MSP Early Finish
Pre-Risk Mitigation	**30 days**	**6/1/20 8:00 AM**	**7/10/20 5:00**		**NA**	**0**
Assembly A	10 days	6/1/20 8:00 AM	6/12/20 5:00	High Risk	NA	0
Assembly B	9 days	6/1/20 8:00 AM	6/11/20 5:00	Low Risk	NA	0
Integration Testing	10 days	6/15/20 8:00 AM	6/26/20 5:00		NA	0
System Test	10 days	6/29/20 8:00 AM	7/10/20 5:00		NA	0
Launch	0 days	7/10/20 5:00 PM	7/10/20 5:00		NA	0

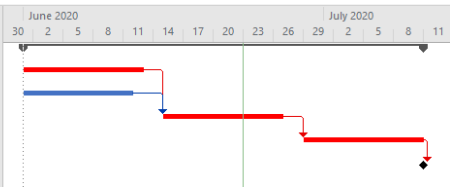


Figure 1

Assembly A and Assembly B are required before Integration Testing can begin. Once Integration Testing is complete there is a period of System testing before a launch milestone of 10Jul20.

The launch window on 10Jul20 must be achieved so we will forecast dates at 100% confidence.

A risk assessment has been performed and Assembly A has been identified as High Risk. This means we have little confidence that Assembly A will be completed by the 12Jun20 date shown in the schedule.

Duration Uncertainty was applied as shown in Figure 2.

ID	Task Name	Remaining Duration	Duration Distribution Type	Duration Optimistic	Duration Most Likely	Duration Pessimistic
0	FM2017 Conditional Branching	30 days	(None)			
1	Pre-Risk Mitigation	30 days	(None)			
2	Assembly A	10 days	Triangular	95%	100%	150%
3	Assembly B	9 days	Triangular	90%	100%	110%
4	Integration Testing	10 days	(None)			
5	System Test	10 days	(None)			
6	Launch	0	(None)			

Figure 2

The pessimistic duration of Assembly A has been set to 150% of the estimated duration. A Risk Analysis is performed and the probability distribution histogram for Launch is shown in Figure 3.

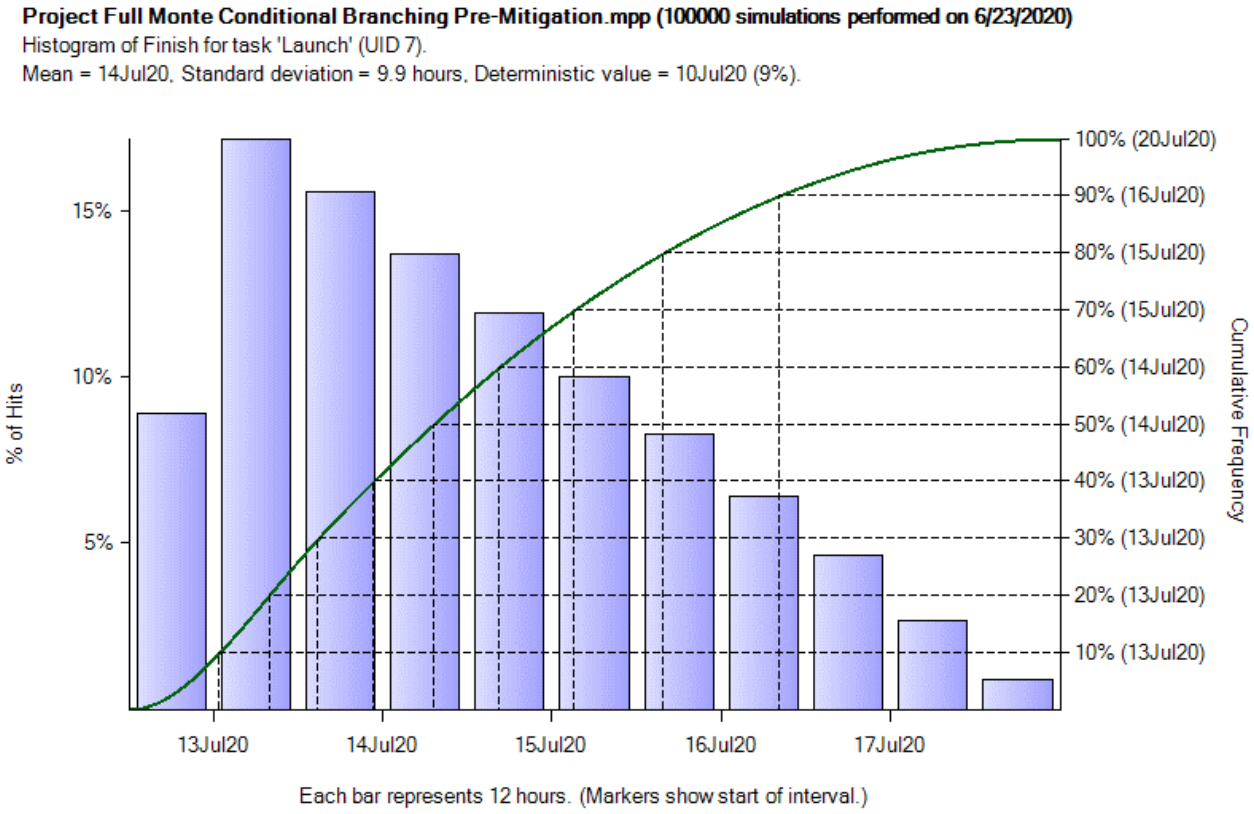
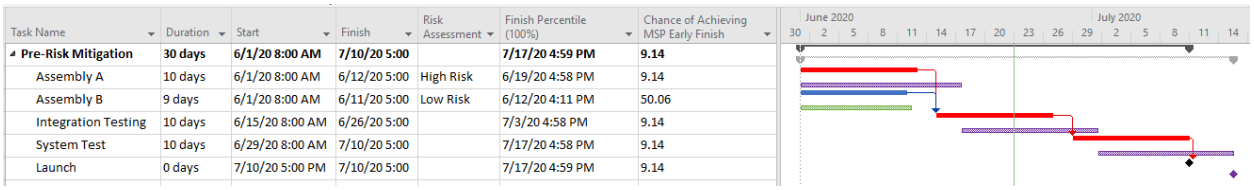


Figure 3

Note: The Histogram does not show the characteristic Normal (Bell Shaped) distribution we typically expect from Schedule Risk Analysis because uncertainty in the schedule is limited to the two Assembly tasks for clarity in the example.

The Schedule Risk Analysis is predicting only a 9% chance of achieving the Launch on 10Jul20. The 100% confidence date is shown as 20Jul20.

Saving the Full Monte results creates the Gantt chart shown in Figure 4.



Task Name	Duration	Start	Finish	Risk Assessment	Finish Percentile (100%)	Chance of Achieving MSP Early Finish
Pre-Risk Mitigation	**30 days**	**6/1/20 8:00 AM**	**7/10/20 5:00**		**7/17/20 4:59 PM**	**9.14**
Assembly A	10 days	6/1/20 8:00 AM	6/12/20 5:00	High Risk	6/19/20 4:58 PM	9.14
Assembly B	9 days	6/1/20 8:00 AM	6/11/20 5:00	Low Risk	6/12/20 4:11 PM	50.06
Integration Testing	10 days	6/15/20 8:00 AM	6/26/20 5:00		7/3/20 4:58 PM	9.14
System Test	10 days	6/29/20 8:00 AM	7/10/20 5:00		7/17/20 4:58 PM	9.14
Launch	0 days	7/10/20 5:00 PM	7/10/20 5:00		7/17/20 4:59 PM	9.14

Figure 4

We can clearly see that the uncertainty applied to Assembly A is delaying subsequent tasks resulting in the unacceptable launch date in over 90% of the simulations. **We have identified a risk!**

Risk Response

The project team will need to decide how to respond to the risk. There are four primary responses to controlling identified risks.

Accept – Decide to take the risk. In this case this is not acceptable as the model suggests a 90% chance of the risk impacting the launch.

Avoid – Change the project strategy so the risk is no longer relevant. No alternatives are available to meet our launch timeframe.

Transfer – Have some third party assume the risk. Unfortunately, nobody will insure our project based on the identified risk.

Mitigate – Take action to reduce the chance of the risk occurring and/or reduce its impact should it occur.

For our example, we are going to **mitigate** the impact of late delivery of Assembly A by specifying alternate logic to employ if the Assembly is delivered late.

If Assembly A is delivered later than 15Jun20, then rather than delay Integration testing, we will perform Additional Unit testing on Assembly A before including it in the final System test. In this case Integration testing will only be performed on Assembly B.

The revised schedule is shown in Figure 5.

Task Name	Duration	Risk Assessment	Start	Finish	Finish Percentile (100%)	Chance of Achieving MSP Early Finish
Post-Risk Mitigation	**30 days**		**6/1/20 8:00 AM**	**7/10/20 5:00**	**NA**	**0**
Assembly A	10 days	High Risk	6/1/20 8:00 AM	6/12/20 5:00	NA	0
Additional Unit Test	5 days		6/15/20 8:00 AM	6/19/20 5:00	NA	0
Assembly B	9 days	Low Risk	6/1/20 8:00 AM	6/11/20 5:00	NA	0
Integration Testing	10 days		6/15/20 8:00 AM	6/26/20 5:00	NA	0
System Test	10 days		6/29/20 8:00 AM	7/10/20 5:00	NA	0
Launch	0 days		7/10/20 5:00 PM	7/10/20 5:00	NA	0

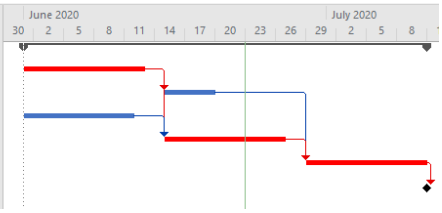


Figure 5

The Additional Unit test has sufficient slack/float to absorb a delayed delivery of Assembly A before inclusion in System Testing.

Post-Mitigation Example

File: Full Monte Conditional Branching Post-Mitigation.mpp

Assembly A now has two successors. The original Finish-to-Start link to Integration Testing and the new Finish-to-Start link to Additional Unit Test. This allows us to implement Conditional Branching.

Launch Full Monte and select task Assembly A. In the right side edit pane, click Branching as shown in Figure 6.

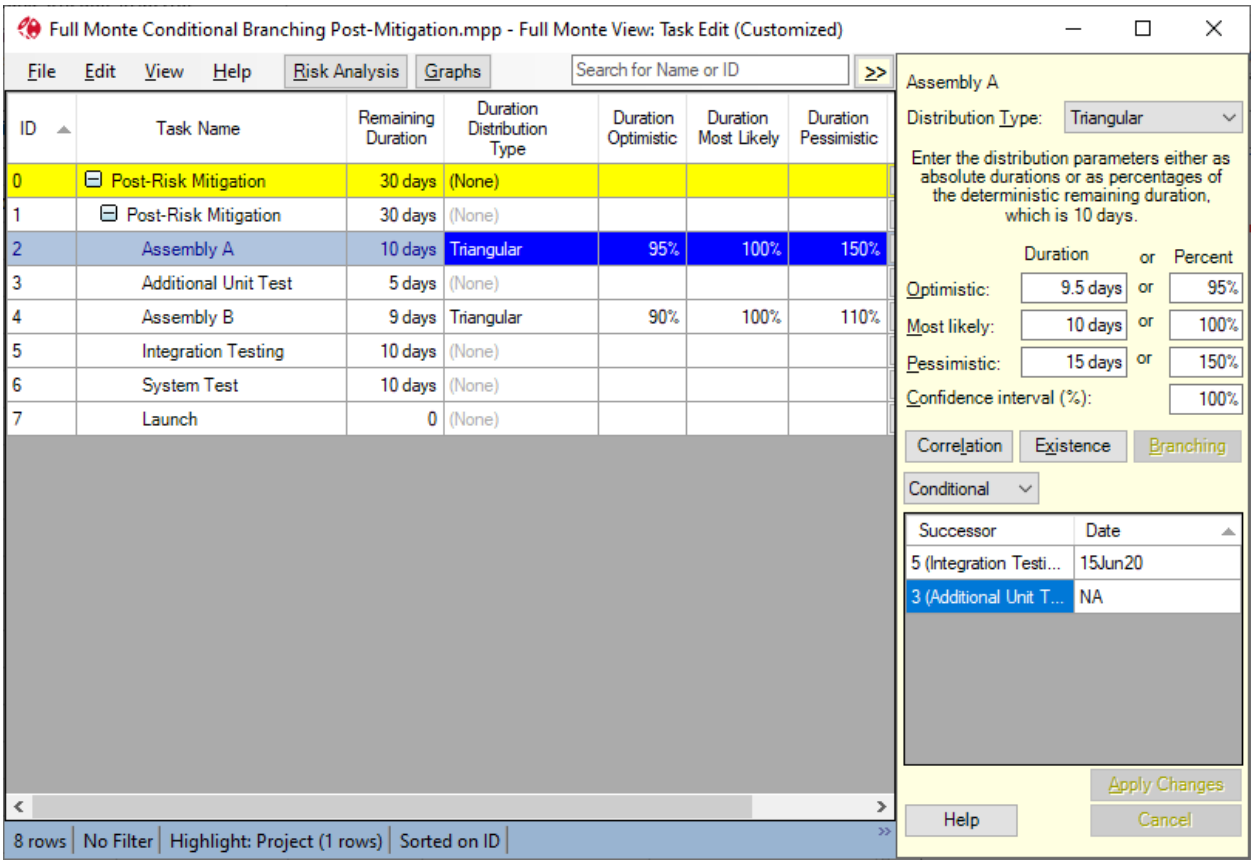


Figure 6

The branching type is set to Conditional and the successors tasks to Assembly A are shown in the grid.

Conditional Branching can be used when a task has two or more successors. The dates in the grid are used to specify which successor should be selected if the finish of Assembly A is on or before the specified date. One successor should be left with no date (NA) and this successor will be executed if the finish of Assembly A is not on or before any of the dates specified for other successors.

In this example, the successor 'Integration Testing' will be executed if the finish of Assembly A is on or before 15Jun20. Otherwise, if the finish of Assembly A is after 15Jun20, the successor 'Additional Unit Test' will be executed. The revised probability distribution histogram for Launch is shown in Figure 7.

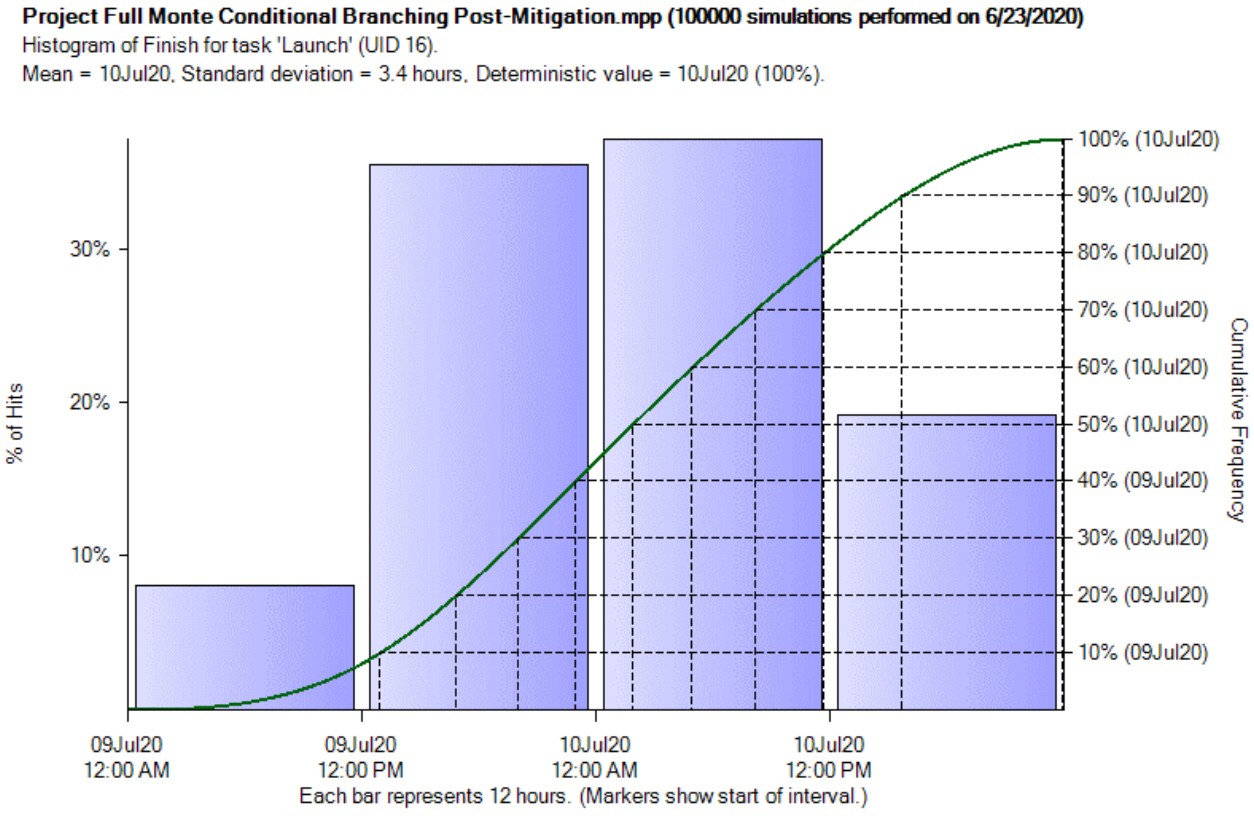


Figure 7

The chance of finishing on 10Jul20 is now 100%.

Saving the results produces the Gantt chart shown in Figure 8.

Task Name	Duration	Risk Assessment	Start	Finish	Finish Percentile (100%)	Chance of Achieving MSP Early Finish
Post-Risk Mitigation	**30 days**		**6/1/20 8:00 AM**	**7/10/20 5:00**	**7/13/20 8:00 AM**	**100**
Assembly A	10 days	High Risk	6/1/20 8:00 AM	6/12/20 5:00	6/19/20 4:58 PM	9.05
Additional Unit Test	5 days		6/15/20 8:00 AM	6/19/20 5:00	6/26/20 4:58 PM	0
Assembly B	9 days	Low Risk	6/1/20 8:00 AM	6/11/20 5:00	6/12/20 4:11 PM	50.16
Integration Testing	10 days		6/15/20 8:00 AM	6/26/20 5:00	6/29/20 8:00 AM	100
System Test	10 days		6/29/20 8:00 AM	7/10/20 5:00	7/13/20 8:00 AM	100
Launch	0 days		7/10/20 5:00 PM	7/10/20 5:00	7/13/20 8:00 AM	100

Figure 8

Note: The Finish Percentile (10%) date shown in the table is 7/13/20 8:00 AM (a Monday). In this specific case, the date is being displayed at the start of the working period immediately after 7/10/20 5:00 PM (a Friday). This is the same instant in time and there is still a 100% chance of launch on 7/10/20 as required.

Conclusion

Conditional Branching allows alternate logic to be followed depending on the finish date of a predecessor task. This is most useful for modelling strategies to mitigate late completion of required work.