Cost Risk Impacts of Schedule Interdependencies

Peter Frederic
Tecolote Research, Inc.

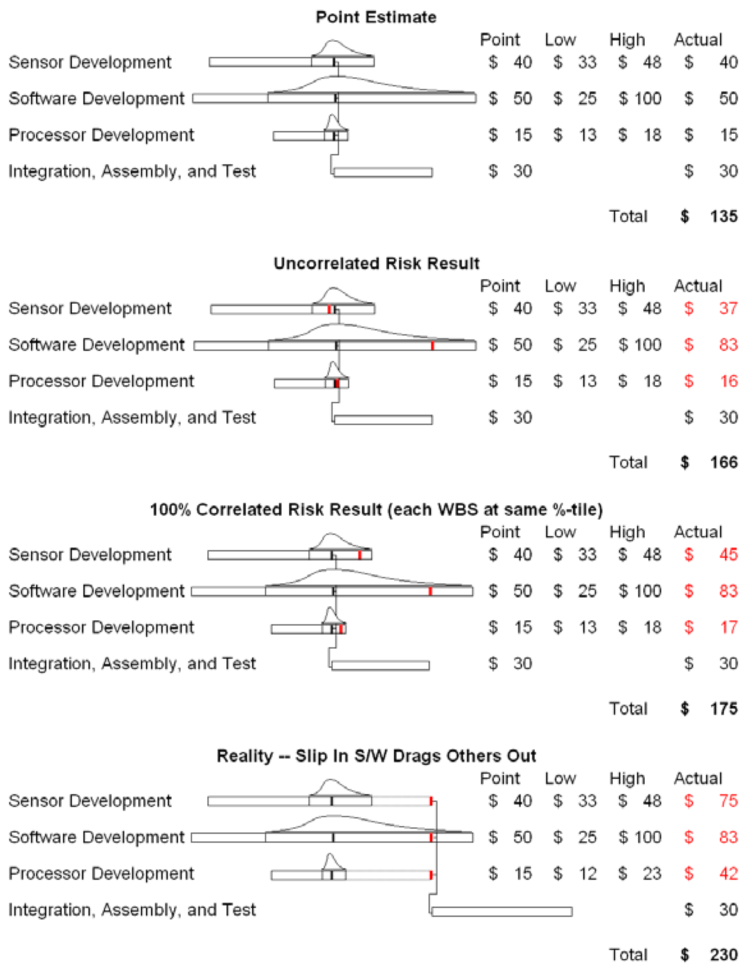
ABSTRACT

This paper introduces an objective methodology for incorporating schedule risk impacts into the cost risk assessment process. Typically, we use penalty factors—in Tecolote Research's ACEIT[*] suite of cost estimating tools—to capture the impact of extraordinary amounts of schedule/technical risk on individual work breakdown structure (WBS) elements. However, this approach is subjective: the technical expert tasked with determining the penalty factor must account for the combined impacts of numerous risk factors. One particular risk factor that is very difficult to characterize is the impact of schedule interdependencies among WBS items. The existence of these interdependencies, particularly in development programs, is widely acknowledged; prominent figures in the cost community advocate the blanket application of correlation factors across all WBS items in a development program to capture the schedule interdependencies. The logic behind this approach is that, in a development program, no WBS item can be completely de-staffed until testing has been successfully completed—therefore, a schedule slip and cost overrun in any one WBS item will cause some degree of overrun in all WBS items. Unfortunately, no degree of assumed correlation between WBS elements can fully capture the potential impact of schedule interactions in a development program.

Consider the example shown below: the top case shows, for a very simple program in ($M), an estimate in which the most likely estimates for each of the WBS items are simply added together (i.e., the point estimate). Think of the remaining cases as iterations in Monte Carlo simulation. In the second case, a value is picked for each item from the specified distribution for that item, but the distributions are assumed to be uncorrelated. The third case is the same as the second, but the distributions are assumed to be completely correlated: since a 90th percentile result occurred for Software, then Sensor and Processor will also experience 90th percentile results. Finally, in the most realistic case, when Software experiences a 90th percentile result, the staff associated with the Sensor and the Processor must be retained many months after their own work would have been completed so that they can support integrated system testing. This example shows how schedule interdependencies can cause an individual WBS item's cost to grow far beyond the cost risk inherent in the item itself.

This paper describes an ACEIT-based methodology to capture a schedule-driven cost impact, such as the one described above, in a direct, functional way. The paper uses a real-world cost model to compare results between the direct approach advocated here and the traditional approach using correlation factors between cost elements. Finally, the paper examines historical development programs in which the traditional, correlation-based approach could not have possibly predicted the cost growth actually experienced.

[*] Automated Cost Estimating Integrated Tools (ACEIT).



Point Estimate

	Point	Low	High	Actual
Sensor Development	$ 40	$ 33	$ 48	$ 40
Software Development	$ 50	$ 25	$ 100	$ 50
Processor Development	$ 15	$ 13	$ 18	$ 15
Integration, Assembly, and Test	$ 30			$ 30
			Total	**$ 135**

Uncorrelated Risk Result

	Point	Low	High	Actual
Sensor Development	$ 40	$ 33	$ 48	$ 37
Software Development	$ 50	$ 25	$ 100	$ 83
Processor Development	$ 15	$ 13	$ 18	$ 16
Integration, Assembly, and Test	$ 30			$ 30
			Total	**$ 166**

100% Correlated Risk Result (each WBS at same %-tile)

	Point	Low	High	Actual
Sensor Development	$ 40	$ 33	$ 48	$ 45
Software Development	$ 50	$ 25	$ 100	$ 83
Processor Development	$ 15	$ 13	$ 18	$ 17
Integration, Assembly, and Test	$ 30			$ 30
			Total	**$ 175**

Reality -- Slip In S/W Drags Others Out

	Point	Low	High	Actual
Sensor Development	$ 40	$ 33	$ 48	$ 75
Software Development	$ 50	$ 25	$ 100	$ 83
Processor Development	$ 15	$ 12	$ 23	$ 42
Integration, Assembly, and Test				$ 30
			Total	**$ 230**

INTRODUCTION

The purpose of this paper is to address a potential weakness in the cost risk assessment process. When a large number of cost elements are added together, if the risk distributions associated with the individual cost elements are totally independent, then the variances associated with the individual elements tend to average out, and the risk distribution of the total cost is actually narrower (in percentage terms) than the individual elements. However, if the cost elements are not totally independent—if there are risk mechanisms that can impact multiple cost elements simultaneously—then this averaging out effect is diminished. Cost risk assessment methodologies that neglect the impacts of interdependencies between cost elements tend to understate risk.

The most important and intuitively obvious mechanisms of interdependency between cost elements are schedule relationships. In complex development programs in particular, there is a strong tendency for delays in the development of one subsystem to cause delays and cost growth in the development of all other subsystems. In the typical development program, the workforce is divided into teams of people for each subsystem. If the system performance requirements are properly allocated to the subsystems, then the interfaces between subsystems are kept as simple as possible, and each team works independently to the largest extent possible. However, eventually all the subsystems must be integrated, assembled, and tested as a complete system. This is when the cost impacts of schedule interdependencies become apparent. Integration activities cannot proceed until all of the subsystems are ready to be integrated. Teams whose subsystems are completed on schedule often have to wait for teams who have experienced delays. It is often impossible for these teams to be "stood down" until testing has been completed. Therefore, significant cost impacts can occur in subsystems that by themselves would be viewed as "low risk."

The impacts of schedule interdependencies on cost risk are widely recognized. Prominent figures in the cost estimating community have advocated the use of statistical correlation factors to capture these impacts. Though all serious risk assessment tools have some means of modeling correlation, unfortunately, in many cases, no degree of correlation can capture the full effect of schedule interdependencies. Consider the example in Figures 1 through 4.

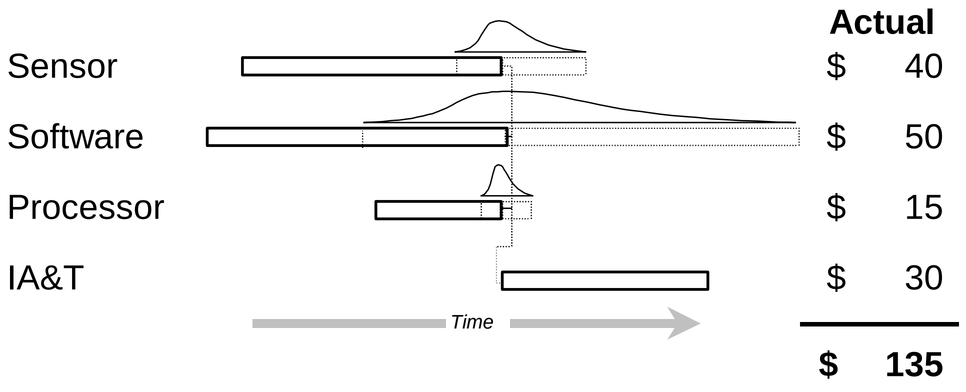


Figure 1: Point Estimate with Schedule Durations and End Date Distributions ($M)

Figure 1 shows, for a very simple program, an estimate in which the most likely estimates for each of the WBS items are simply added together (i.e., the point estimate). The horizontal bars in the middle of the figure represent the scheduled duration for the development of each subsystem. The bell shapes over each bar represent the probability density function for the finish date (and cost) of each subsystem—in other words, the risk distribution. The final bar represents the Integration, Assembly, and Test (IA&T) activity. The vertical lines joining the bars represent scheduling precedence relationships: the IA&T activity cannot start until development of each subsystem is complete. The total of $135M is the sum of the "most likely" estimates for each subsystem and IA&T. This total is a number with no particular statistical significance: it is useful mainly to validate the behavior of the cost model.

To obtain a cost estimate with a known level of confidence, one might use Monte Carlo simulation. In Monte Carlo simulation, the cost model is recalculated iteratively, picking different values from the distribution of possible values for each cost element in each iteration. Figure 2 represents a single iteration in a Monte Carlo simulation. In Figure 2, a value is picked for each item from the specified distribution for that item, but the distributions are assumed to be uncorrelated. The red lines represent the finish dates picked for each subsystem in this iteration. The new cost values correspond directly to the new finish dates. Notice that the cost of the IA&T activity has also changed: in this example, IA&T is calculated using a factor applied to the sum of the subsystem costs. This is often referred to as "functional correlation." In this iteration, a particularly high-cost outcome for the Software subsystem has been somewhat offset by relatively low costs for the Sensor and Processor subsystems. But is this a realistic outcome? Surely the fact that the Software took 67% longer to develop than originally planned would have some impact on the other components of the program.

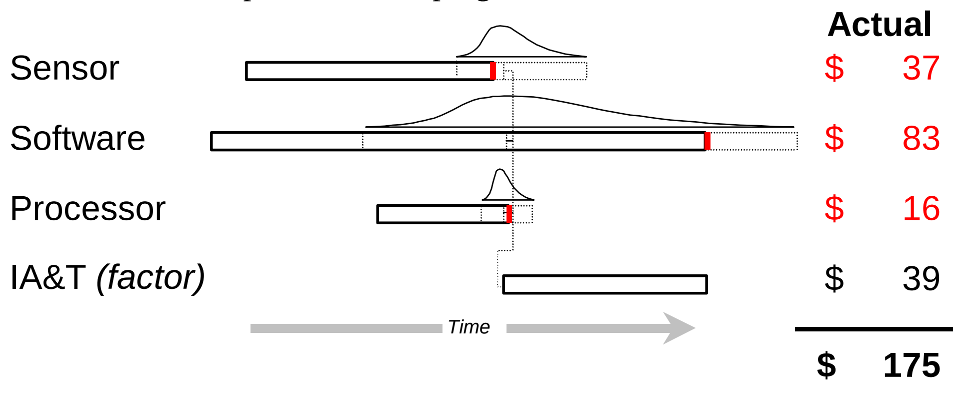


Figure 2: Monte Carlo Result, No Correlation ($M)

In Figure 3, we consider the same set of outcomes for each subsystem. However, we attempt to use statistical correlation to address the impact of the Software disaster on the rest of the project. The distributions are assumed to be completely correlated, meaning that, since a 90th percentile result occurred for software, then Sensor and Processor would also experience 90th percentile results. However, because we have no explicit way of recalculating the schedule dates in our cost model, we have no way of knowing that the IA&T activity will actually slip by several months. By relying solely on statistical correlation, we have effectively assumed that IA&T activity will proceed on the originally planned start date regardless of the fact that the Software doesn't exist yet and that we are not accounting for the whereabouts of the Sensor and Processor teams in the time between completion of their individual efforts and completion of the Software.

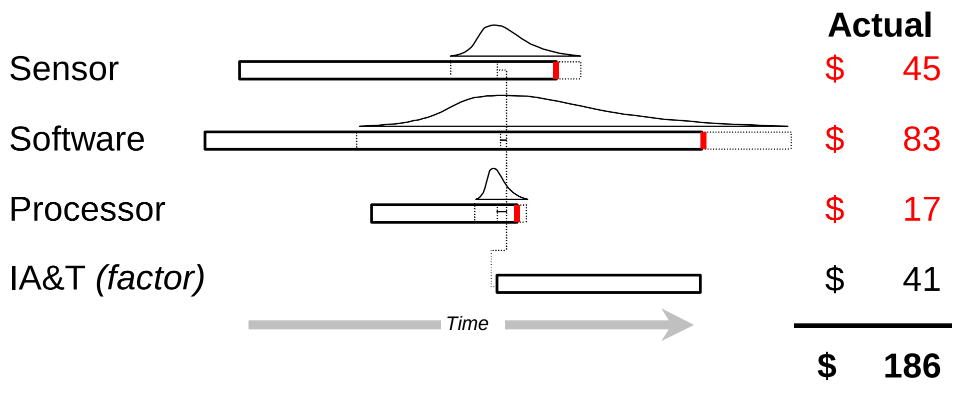


Figure 3: Monte Carlo Result, 100% Correlation ($M)

Finally, in Figure 4, the most realistic case, we have enforced the scheduling precedence relationships between the subsystem efforts. When Software experiences a 90th percentile result, the staff associated with the Sensor and the Processor must be retained many months after their own work would have been completed so that they can support integrated system testing.

Furthermore, the IA&T effort has been increased to reflect the fact that personnel and facilities that had been reserved starting at the original planned start date for IA&T sat idle until the subsystems were actually available for integration. The resulting total cost is 34% higher than when we applied 100% correlation!

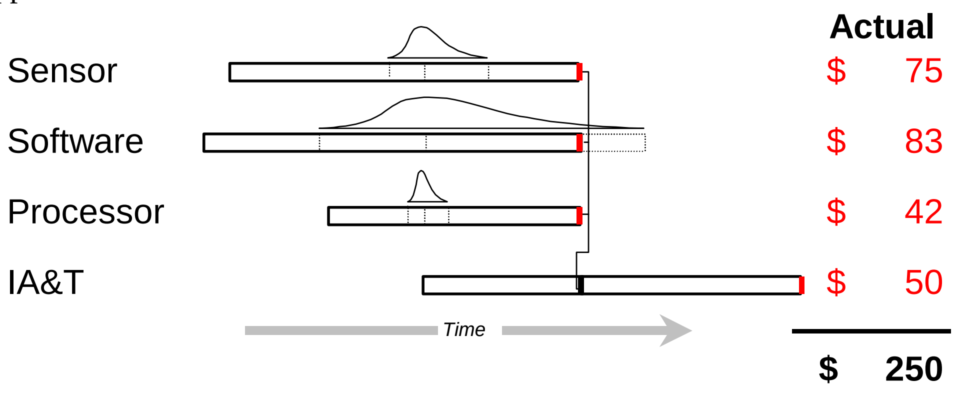


Figure 4: Monte Carlo Result, Schedule Links Enforced ($M)

This example shows how schedule interdependencies could cause an individual WBS item's cost to grow far beyond the cost risk inherent in the item itself, and even beyond what we could model using the most aggressive assumptions about correlation between subsystem costs.

One important feature of this example is that there is one cost element with significantly more risk than many of the other elements. If every subsystem had the same amount of risk (in schedule terms), then 100% correlation would result in the same schedule extension for each subsystem, which would have the same effect as directly enforcing the scheduling precedence relationships. However, by definition, all development programs have some new features, so there are almost always one or more subsystems that are more risky than the others.

ISSUES

The astute reader may have noticed a few key assumptions embedded in the example above. First, the example assumes a one-for-one relationship between schedule duration and cost. This is only realistic for cost elements that are composed purely of labor costs. Cost elements that include a significant material cost component (or purchased parts, or fixed-price subcontracts) will exhibit a less than one-for-one relationship between schedule and cost. Schedule duration growth will only impact the labor portion of the cost. For example, if a cost element consists of 50% labor and 50% material, and the schedule duration grows by 100%, the overall cost will increase by only 50%. On the other hand, if we wish to infer a schedule slip based on cost growth, the effect is reversed: a 50% cost growth in a cost element that consists of only 50% labor implies 100% duration growth.

In addition to the non-labor contents of most cost elements, there are other factors that may impact the relationship between cost and schedule. If a schedule slip is anticipated and planned for, it may be possible to reduce staffing on the subsystem teams that are *not* behind schedule to reduce costs while waiting for the late subsystems to catch up. On the other hand, it might be possible to increase the staffing on a subsystem that is running late to prevent a slip in the overall schedule that would cause cost growth in other subsystems.

Another important assumption is that the baseline cost estimate and the baseline schedule are actually related somehow. This seems like an obvious assumption, but is not always true. Often, program schedules and cost estimates are prepared by separate camps. If the schedule and the cost estimate cannot be related to each other through a realistic staffing plan, then any attempts to relate schedule growth to cost growth will be meaningless. This may explain why efforts to uncover statistical relationships between schedule slips and cost growth in Selected Acquisition Report (SAR) data have been frustrating.

The remainder of this paper will describe a method for implementing the approach illustrated in Figure 4 using the ACEIT cost-modeling tool.

IMPLEMENTATION ALGORITHM

To explicitly model the cost impacts of schedule interactions using Monte Carlo simulation, the following calculation steps can be performed in each iteration:

1. Infer schedule slips based on cost growth for each WBS item
2. Recalculate schedule milestones based on slipped durations and logical precedence relationships
3. Recalculate costs based on slipped schedule milestones

Each of these steps is explained in detail in the following sections.

Inferring Schedule Slips

The first step in calculating the cost risk impact of schedule interactions is to translate cost growths for individual cost elements into equivalent schedule duration growth. The rationale behind this operation is that duration growth is equivalent to effort growth, which in turn is equivalent to cost growth. We assume that the cost growth for an individual cost element includes the impact of "Cost Estimating Relationship (CER) Risk," input variable risk, and technical risk, but not system-level schedule risk impacts. In other words, we assume that systems with known schedule anomalies have been eliminated from CER databases.

The general formula for calculating duration growth for an individual cost element is

Duration Growth Factor	=	1.0 + (Cost with Risk/Cost without Risk - 1.0) * Schedule-Related Cost Factor

where

Cost with Risk	=	the cost selected from the risk distribution for the cost element in a particular Monte Carlo iteration
Cost without Risk	=	the point estimate for the cost element

Schedule-Related Cost Factor	=	a factor indicating the portion of the cost element that is actually related to duration

Nominally, the Schedule-Related Cost Factor would be the inverse of the material cost percentage for the cost element. However, as we will see in subsequent sections, the opposite effect occurs when we translate schedule slips back into cost impacts, so it is convenient to simply leave this factor set to 1.0. In addition to the non-labor content of cost elements, there may be other effects that would impact the Schedule-Related Cost Factor. For instance, if management recognized that a particular cost element/team/subsystem was running behind schedule, they might increase staffing on that team or authorize overtime to prevent further schedule impacts. This effect is hard to quantify, but the Schedule-Related Cost Factor provides a mechanism to incorporate it if needed.

In ACEIT, the point estimate values for the "Cost without Risk" variable are not directly accessible during Monte Carlo iteration. It is necessary to manually copy the point estimate into a separate Dynamic Equation Column (DEC) prior to the risk run. These values can then be accessed during Monte Carlo iteration. The following is an example of how the duration growth factor would appear in an actual ACEIT equation.

```
(1.0 + (FYTOT(@EngDlrPCS)/EngDlrPCS.NoRisk - 1.0) * SGFactor)
```

Recalculating Schedule Milestones

Once the individual schedule duration impacts have been determined, we need to determine the impact to the overall program schedule. To do this in each Monte Carlo iteration requires that a dynamic model of the program schedule be built into the cost model. This is not as complicated as it sounds. The goal is mainly to capture the slips or (advances) in key program-level milestones. Most development programs proceed according to the following fundamental sequence:

1. **Preliminary Design:** System-level performance requirements are allocated or "flowed down" to subsystems. Interfaces between subsystems are defined. For each subsystem, this activity culminates in a subsystem Preliminary Design Review (PDR) in which the subsystem requirements are reviewed for internal consistency.
2. **System PDR:** The subsystem preliminary designs are reviewed to make sure they are consistent with each other and that all system-level performance requirements have been addressed. System PDR cannot be truly completed until all subsystem PDRs have been successfully completed. Unfortunately, this rule is often bent.
3. **Detailed Design:** The subsystem performance requirements are translated into operational designs. This often requires fabrication of engineering models and experimentation to determine if design choices are correct and verify the predictions of engineering models. The result of this activity is a set of detailed engineering drawings for each subsystem. This activity culminates in a subsystem Critical Design Review (CDR) in which the subsystem requirements are reviewed for internal consistency.

4. **System CDR:** The subsystem detailed designs are reviewed to make sure they are consistent with each other. System CDR cannot be truly completed until all subsystem CDRs have been successfully completed. Again, this rule is often bent.
5. **Fabrication, Assembly, and Unit Test:** Subsystem operational prototypes are manufactured and unit tested. Software coding and unit test also occurs in this phase of the program.
6. **System Integration, Assembly, and Test (IA&T):** Once the subsystem prototypes have been fabricated and individually tested, they must be assembled and tested as a complete system. Needless to say, system testing cannot be completed until all subsystems have been completed. However, for low volume, highly complex systems such as spacecraft, the delivery of subsystems does not necessarily have to be simultaneous. Some systems may not be required until later in the assembly sequence.

The System PDR, System CDR, and System IA&T are the principal points in the program where a slip (duration growth) in any one subsystem can cause the overall program to be delayed and therefore cause cost growth in all subsystems. To capture these effects, the schedule has to be modeled to at least the level of detail shown above. Lower levels of detail are unnecessary: at this level of detail, most of the sophisticated modeling features of dedicated schedule modeling tools such as Microsoft Project provide no additional utility. The schedule can be modeled at this level with simple finish-to-start precedence relationships. These are easily implemented using the MAX function in ACEIT. The general formula for modeling these relationships is

Activity Finish Date = MAX(P1 Finish, P2 Finish, P3 Finish, … Pn Finish) + Duration * Duration Growth Factor

where

Px Finish	=	the finish date of each predecessor activity
Duration	=	the duration of the schedule activity whose finish date we are calculating
Duration Growth Factor	=	the duration growth factor for this activity as calculated in the previous step ("*Inferring Schedule Slips*")

In some cases, even the simple schedule that we must model may be slightly more detailed than the cost element level of detail at which we calculate the Duration Growth Factors. For example, we might use a CER for a Structural subsystem that estimates total non-recurring cost, including both Preliminary Design and Detailed Design. In this case, we would only be able to calculate one Duration Growth Factor. However, we can simply apply this same factor to both the Preliminary Design activity and the Detailed Design activities, effectively assuming that the duration growth will be spread evenly over the span of Preliminary Design and Detailed Design.

Recalculating Costs Based on Schedule Slips

Once we have calculated an updated schedule incorporating the impacts of risk, we can go back through the cost elements and calculate the cost impact on each cost element. Each element is assumed to have a baseline expenditure profile that is based on the point estimate for that element and the initial planned schedule. As the major milestones to which the expenditure profile is tied slip to the right, the expenditure profile stretches—and cost increases. A special User Defined Function (UDF) called “Stretch” was developed in ACEIT to model this expenditure stretching effect. This is illustrated in Figure 5.

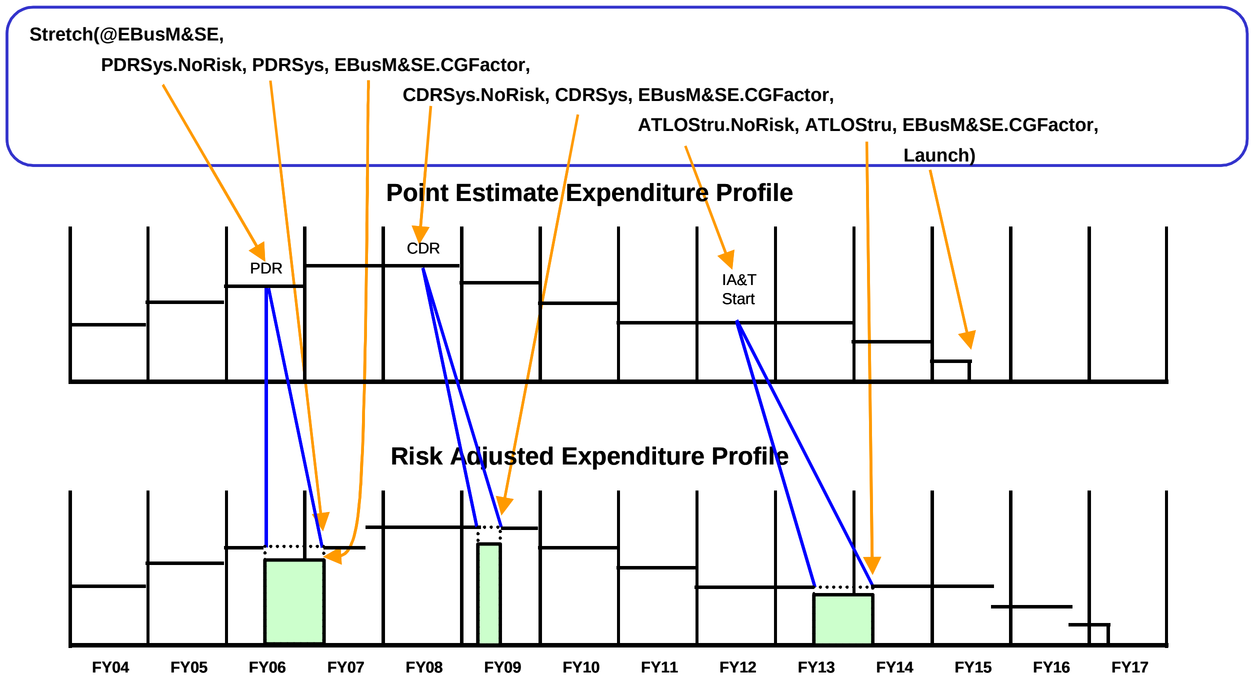

Stretch(@EBusM&SE,
PDRSys.NoRisk, PDRSys, EBusM&SE.CGFactor,
CDRSys.NoRisk, CDRSys, EBusM&SE.CGFactor,
ATLOStru.NoRisk, ATLOStru, EBusM&SE.CGFactor,
Launch)
Point Estimate Expenditure Profile
PDR
CDR
IA&T Start
Risk Adjusted Expenditure Profile
FY04
FY05
FY06
FY07
FY08
FY09
FY10
FY11
FY12
FY13
FY14
FY15
FY16
FY17

Figure 5: Application of Funding Stretch Function

Figure 5 shows an application of the new Stretch UDF. The Stretch function allows an expenditure profile to be stretched (or contracted) at as many as three milestones. The first parameter in the Stretch call above refers to the ACEIT row containing the time-phased point estimate—“EbusM&SE” in this case. The next three parameters describe the stretch to be applied at the first milestone. In the example, the first milestone is the system PDR. The “PDRSys.NoRisk” parameter refers to a row containing the original planned PDR date. The “.NoRisk” notation tells ACEIT to get the value from a Dynamic Equation Column called “NoRisk” into which all values from the original point estimate have been previously copied and pasted. The “PDRSys” parameter (*without* the “.NoRisk” notation) refers to the same row, but refers to the active calculation column that contains the current calculated value for the PDR date, with the risk impact calculated for the current Monte Carlo iteration. As shown in the figure, “PDRSys.NoRisk” is the stretch-from date, and “PDRSys” is the stretch-to date. The “EbusM&SE.CGFactor” parameter is a factor that can be used to specify a less than one-for-one relationship between duration growth and cost growth. Specifying a number less than 1.0 effectively reduces the burn rate during the stretch period. As mentioned in the “Inferring Schedule Slips” section above, a starting point for this parameter might be the inverse of the material cost percentage for the cost element. However, since this factor would be counteracted by the opposite effect in cost growth-to-duration growth relationship (see inferring schedule slips), we simply leave this factor set to 1.0.

The next row of three parameters describes the stretch to be applied at the CDR milestone similarly to the way that the PDR stretch was applied. The following row describes the stretch to be applied to the final milestone, which is when the subsystem in question ("Structure" in the example) has completed integration in the Assembly, Test, and Launch Operations (ATLO) sequence.

The final parameter is the date that marks the end of the expenditure profile for the cost element. In the example, development expenditures are assumed to end on the day that the spacecraft is launched.

Figure 5 reflects the fact that from the user's perspective, ACEIT tracks cost on a fiscal-year-by-fiscal-year basis. This causes the Stretch UDF to be quite complicated internally, because it has to be able to account for schedule slips and contractions that are much less than a year long. However, if the Stretch UDF is implemented as a built-in ACEIT function, much of this complexity will be eliminated because ACEIT can internally calculate expenditures on a day-by-day basis.

REAL-WORLD EXAMPLE

To test our direct approach to modeling schedule interactions during risk assessment, we have applied it to an actual cost estimate that is in progress at Tecolote. The estimate that we are using as a test case is for the development, including first flight, of an advanced space vehicle. **All of the cost numbers and technical parameters in the figures in this section have been adjusted up or down by significant amounts to protect the identity of the system.** Additionally, key WBS items and schedule activities have been renamed.

Figure 6 shows the estimating WBS for the example system. It is a fairly typical space vehicle development program. Where zeros appear or WBS items might appear to be missing, this is because this is an actual estimate in progress, and there are still a few boxes that have yet to be filled in. However, the estimate is more than complete enough to demonstrate the impact of explicitly modeling schedule interactions versus using correlation factors in the risk assessment process.

For each hardware subsystem in the WBS, there are separate elements for Engineering (or nonrecurring) costs and First Flight Unit (or recurring) costs. For selected subsystems, there are also Technology Development cost elements, though most of these are grouped under "Government Spacecraft Module."

WBS/CES Description	**BASELINE**
** Final Cost by WBS WITH Schedule Impact	
Total System DDT&E	$ 8,409.19 *
Space System	$ 8,409.19 *
Space System Mgmt & Sys Engr	$ 0.00 *
Solar Power Module	$ 3,451.44 *
Technology Development	$ 1,074.47 *
Engineering	$ 2,310.68 *
First Flight Unit	$ 66.29 *
Spacecraft Module	$ 3,931.58 *
S/C Module Mgmt & System Engr	$ 744.53 *
Spacecraft Module Mgmt	$ 102.71 *
Spacecraft Module System Engineering	$ 102.71 *
Spacecraft Module Assurance	$ 308.12 *
Power Converter System Design	$ 231.00 *
Power Conversion Segment	$ 621.80 *
Engineering	$ 390.00 *
First Flight Unit	$ 231.80 *
Propulsion Segment	$ 108.35 *
Engineering	$ 60.72 *
First Flight	$ 47.63 *
Bus Segment	$ 827.08 *
Bus Management and System Engineering	$ 214.78 *
Engineering	$ 138.01 *
First Flight	$ 76.77 *
Attitude & Articulation Control Subsystem	$ 145.23 *
Engineering	$ 88.25 *
First Flight Unit	$ 56.98 *
Command and Data Handling Subsystem	$ 27.10 *
Engineering	$ 14.48 *
First Flight Unit	$ 12.62 *
Environmental Monitoring Subsystem	$ 27.30 *
Engineering	$ 20.60 *
First Flight Unit	$ 6.70 *
Bus Assembly, Integration & Test	$ 78.41 *
Engineering	$ 55.19 *
First Flight Unit	$ 23.22 *
Launch Vehicle Adapter Segment	$ 15.91 *
Engineering	$ 14.17 *
First Flight Unit	$ 1.74 *
Spacecraft Module Assembly, Integration and Test	$ 242.61 *
Engineering	$ 199.95 *
First Flight Unit	$ 42.65 *
Spacecraft Module Test and Evaluation	$ 553.72 *
Spacecraft Module System Level Software Segment	$ 441.78 *
Spacecraft Module Test Facilities	$ 0.00 *
Gov't Spacecraft Module	$ 375.81 *
Gov't Rad Hard Technology	$ 0.00 *
Govt' Power Conversion	$ 301.09 *
Gov't Propulsion	$ 74.72 *
Gov't Telecom	$ 0.00 *
Gov't Optical Nav Camera Subsystem	$ 0.00 *
Gov't S/C Self Inspection Camera Subsystem	$ 0.00 *
Mission Module	$ 1,026.17 *
Engineering	$ 868.79 *
First Flight	$ 157.38 *

Figure 6: Estimating WBS

Figure 7 is a Gantt chart representing the planned schedule for the example program.

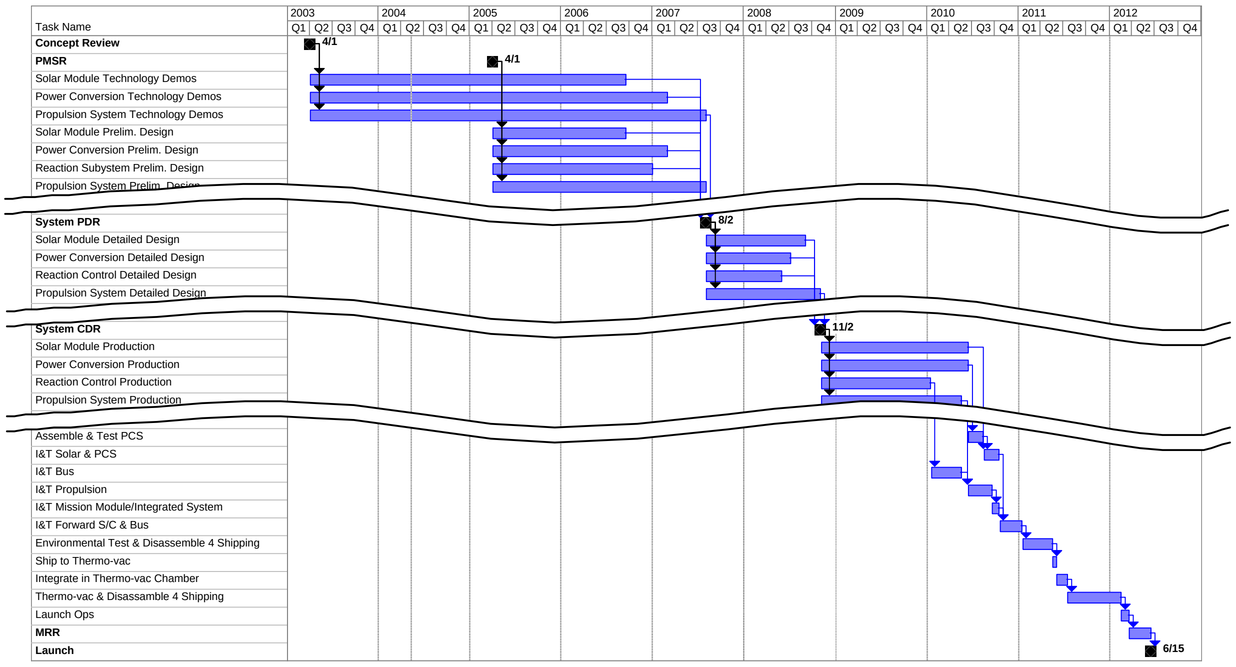


Figure 7: Schedule Model

This schedule follows very closely the sequence described above in “Recalculating Schedule Milestones”. The only significant exception is the addition of Technology Demonstration activities that precede the System PDR.

Figure 8 shows the results of applying Monte Carlo simulation in ACEIT to the example cost and schedule model. The different S-Curves represent different approaches and assumptions used to model schedule interactions. In all cases, the risk distribution assignments for the individual cost elements were the same:

Table 1: Cost Element Risk Distributions for Example Space System

Cost Type	Distribution Form	Low % (Relative to Point)	Low Interpretation (%-tile)	High % (Relative to Point)	High Interpretation (%-tile)
Technology Development	LogNormal	50%	5	200%	95
Engineering	LogNormal	66.67%	5	150%	95
First Flight Unit	LogNormal	80%	5	125%	95
Exceptions:					
Power Conversion Technology Development	LogNormal	33%	5	300%	95
Propulsion Technology Development	LogNormal	33%	5	300%	95

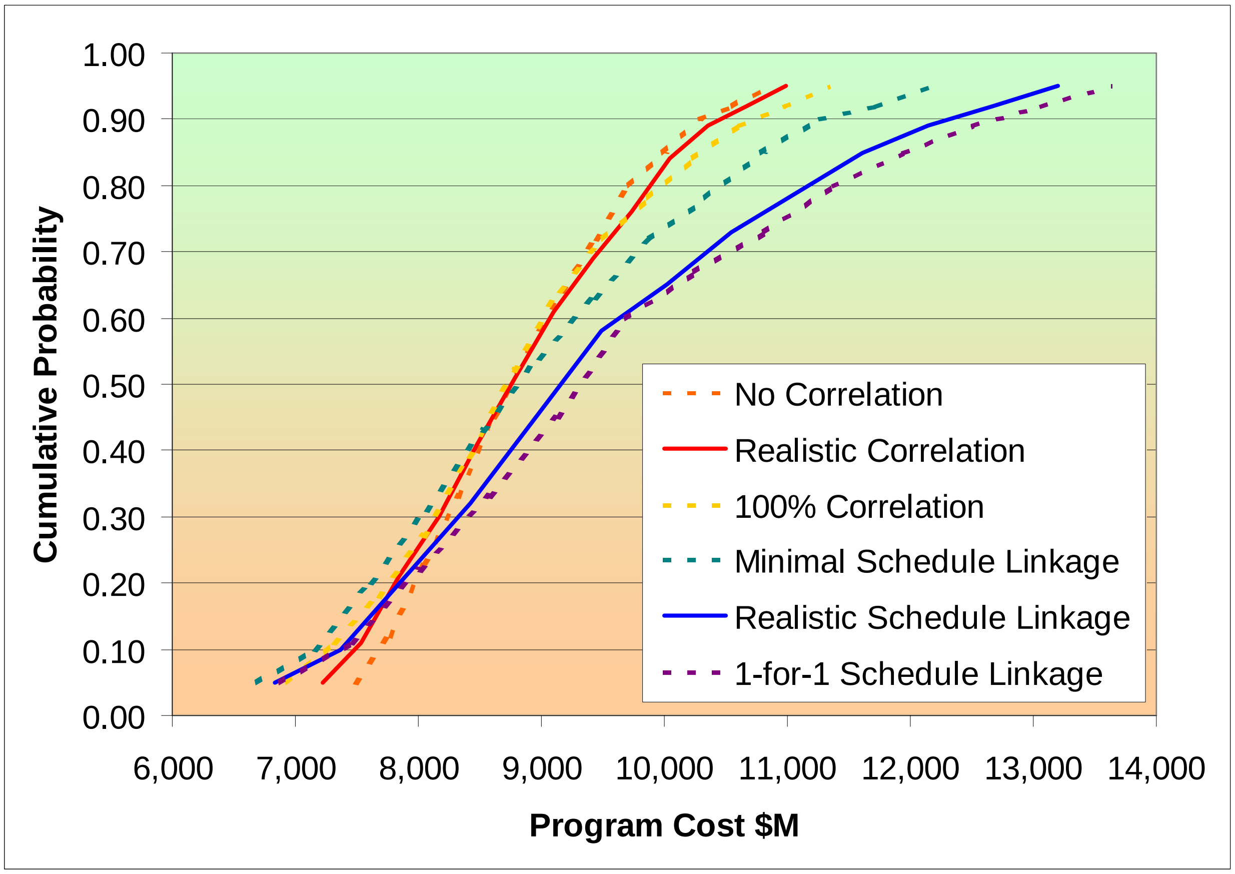


Figure 8: Monte Carlo Results for Different Methods of Modeling Schedule Interactions

In Figure 8, the "No Correlation" curve represents the results if the model is run with no correlation (known as "Grouping" in ACEIT terminology) specified for any cost element and with explicit schedule interrelationships not enforced. For the "Realistic Correlation" curve, ACEIT's correlation capability is used with intuitively reasonable correlation factors: 0.1 between all Technology Development cost elements, 0.8 between all Engineering cost elements, and 0.2 between First Flight Unit cost elements. For the "100% Correlation" curve, the correlation factors are 1.0 between all Technology Development cost elements, 1.0 between all Engineering cost elements, and 1.0 between First Flight Unit cost elements. For the "Minimal Schedule Linkage" curve, no correlation is specified, but schedule interdependencies are enforced. All cost growth factors used in the Stretch function (see Recalculating Costs Based on Schedule Slips) are set to 0.3. The "Realistic Schedule Linkage" curve is the same as "Minimal Schedule Linkage," except the cost growth factors used in the Stretch function are set to 0.8. In "1-for-1 Schedule Linkage," the cost growth factors used in the Stretch function are set to 1.0.

The results show a significant but not outlandish increase in the cost risk identified in the development program using the approach of modeling scheduling interactions directly versus using statistical correlation to mimic the effects of schedule interactions. Specifically, the 80th percentile cost produced by the "Realistic Schedule Linkage" model is 12% higher than the 80th percentile estimate produced by the "100% Correlation" model.

HISTORICAL COST GROWTH DATA

One of the objectives of this research effort was to analyze historical cost growth data to uncover statistical evidence to support the relationship between schedule growth and cost growth. However, a quick literature search revealed that this analysis had already been done. A carefully crafted paper presented at the 2002 SCEA Conference, titled "Schedule and Cost Growth" by Coleman, Summerville, and Dameron [1] examined that issue in great detail, and discovered no statistic evidence of any relationship between cost growth and schedule growth! Although every statistical avenue appears to have been investigated, this result does not square with conventional wisdom. When programmatic problems are reported in aerospace trade journals such as *Aviation Week* and *Space News*, the terms "cost overrun" and "schedule slip" are used synonymously. Any aerospace engineer or manager will agree that the longer it takes to do something, the more it will cost.

The Coleman study was based on data from Selected Acquisition Reports (SARs). One possible explanation for the lack of any observable relationship between cost growth and schedule growth in this data is that much of the schedule growth in the database may have been caused by budget constraints. Although most program managers would argue that budget constraints cause overall program costs to go up, this increase is less than if the schedule growth was caused by lack of technical progress. For example, if a program's schedule is based on spending $1M in one year, but only $500K is available in that year and $500K in the next year, the program will take two years to complete. However, overall cost will not necessarily grow. In this case, we have had a *planned* schedule slip, achieved by reducing the expenditure rate. On the other hand, if the same program was not subject to any funding constraint, but due to technical shortfalls, we found ourselves only half way done at the end of the first year, the project would take two years to complete, and would cost twice as much. This would be an *unplanned* schedule slip. The cost/schedule risk assessment approach described in this paper deals with the cost impacts of *unplanned* schedule slips only.

Another possible explanation is that the baseline schedule and baseline cost estimate for a program may not be strongly tied. Typically schedules and cost estimates are developed by separate teams. The initial schedules for most programs are determined by starting from the perceived "need date" and working backwards. Initial program cost estimates are usually developed using parametric CERs that are not often sensitive to schedule milestones. Because there is often no hard logical tie between the schedule and the cost estimate, there is no reason to expect the error in the cost estimate to be correlated with the error in the schedule. The approach described in this paper is predicated on the assumption that the cost estimate and the schedule have been developed jointly, or at least have been reconciled. It is critical that the amount of effort identified in the cost estimate can be executed within the constraints of the schedule, given a reasonable staffing profile.

Although the SAR data provides no strong evidence to support a relationship between cost growth and schedule growth, neither does it prove that such a relationship does not exist given the preconditions stated above: we only address unplanned slips and we assume that the schedule and the cost estimate are actually logically related to each other.

CONCLUSIONS

We have shown that modeling schedule interactions directly is a much more powerful method than statistical correlation for assessing the cost risk impacts of these interactions. It is also more intuitively pleasing. However, this does not diminish the usefulness of addressing correlations in risk assessment tools: there are other effects besides schedule interdependencies that can be efficiently addressed using correlation—hardware commonality and engineering relationships between technical variables are two examples.

Finally, we have shown that the direct approach to modeling schedule interactions can be implemented without adding an unreasonable amount of complication to an ACEIT cost model. It should be noted though that use of the Stretch UDF seems to have a significant impact on computation time when running Monte Carlo simulation. However, it is probable that if this function was implemented as a built-in function in ACEIT, the speed penalty could be eliminated.

REFERENCE

1. “Schedule and Cost Growth,” R. L. Coleman, J. R. Summerville, M. E. Dameron, Northrop-Grumman/TASC, SCEA National Conference, (Phoenix), 11-14 June 2002.