LINKING PARAMETRIC ESTIMATES TO PROGRAM MANAGEMENT ARTIFACTS (LPEPM)

Ted Mills, NASA Cost Analysis Division; Mike Smith, Booz Allen Hamilton; John Swaren, PRICE Systems

Abstract

A common fate of parametric cost and schedule estimates is that they fall into disuse as a Project's own artifacts (e.g. Work Breakdown Structure (WBS), budget, schedule, risk lists, etc.) are created and mature. Parametric estimates typically do not map cleanly to WBS or schedule-derived artifacts, allowing a sense among Project Managers (PMs) – rightly or wrongly – that "parametric estimates are fine, but they don't reflect my project." As a result of this bias, parametric estimates and the estimators that generate them find themselves relegated to obscurity after passing the first project milestone. The problem lies in that dismissing parametric estimates on these grounds, PMs lose the benefit of the historic realities captured in Cost Estimating Relationships (CERs) that drive the models. Conversely, cost estimators have observed that the recent Joint Cost-Schedule Confidence Level (JCL) analyses required by NASA policy to occur at PDR/KDP-B, have yielded suspiciously narrow Coefficients of Variation in JCL cost S-curves. This observation gives rise to concerns within the cost community that projects, overly reliant on their own SMEs to provide uncertainty ranges, are missing opportunities to incorporate significant uncertainties into their estimates.

NASA's Cost Analysis Division (CAD), Booz Allen Hamilton and PRICE Systems collaborated to conduct research into linking parametric estimates to programmatic artifacts in a manner that would elevate parametric estimates and allow Programs and Projects to apply the historical lessons that make parametric estimates so powerful and accurate. This research brought together parametric and programmatic cost estimators, model developers, software developers, schedulers, risk analysts and practitioners ranging from junior analysts to Ph.D. thought-leaders to think through and articulate a process by which parametric cost estimates could be linked to programmatic artifacts in a manner that takes maximum advantage of the best each has to offer. Specifically, the collaborative research evaluated the feasibility of a parametric cost model "informing" a JCL model and vice-versa via iterative methodology. This research resulted in a practical, clearly-articulated process for performing this cross-informing linkage, as well as the development of standardized enabling tools (data collection templates and a dashboard tool) through which to visualize and perform iterative comparative analyses. The research used as a test-case a contemporary, real-world NASA project which needed only to meet two conditions: that a recent parametric estimate have been performed; and it had been through a JCL analysis. This ensured that a requisite set of comparable programmatic and parametric products existed. With those paired data sets, the LPEPM research team deconstructed the models and developed a process for linking parametrics to programmatic artifacts and proved that the concept can be executed and has merit. The team encountered challenges resulting in lessons-learned designed to benefit any analyst in the field attempting such a linkage.

Introduction

Currently NASA is using a variety of cost analysis methodologies to formulate, plan, and implement projects.

In the formulation stage, specifically for KDP-B[1], NASA is calling for programs and projects to provide probabilistic analysis on both their cost and schedule estimates. This analysis is then used to determine a high and a low estimate for cost, and for schedule. The community has identified two candidate methodologies for producing the risk estimates and associated results: 1) complete parametric estimates of cost and schedule or, 2) complete a Joint Cost-Schedule Confidence Level (JCL) analysis. Conducting a JCL at KDP-B is not normally required. This is primarily because projects typically do not have detailed plans available to support an in-depth JCL analysis, and by design, the requirement at KDP-B is intended to "bound the problem." Conducting a parametric estimate of schedule and cost utilizes the historical data and performance of the Agency and provides a valuable estimate of the range of possibilities.

The parametric analysis done at KDP-B is very valuable due to several reasons. First, the parametric analysis utilizes technical information (independent variables) and "correlates" them to programmatic information (dependent variables) such as cost and schedule. Second, parametric analysis, as implied above, provides a statistical rigor to estimating the amount of effort (typically quantified in terms of cost and schedule). Parametric analysis will generally do a good job of estimating unless:

a) Project has major cost drivers not modeled by the parametric model
b) The parametric model does not contain data analogous to estimate
c) The programmatic content and approach are not defined properly (test hardware, development approach, funding availability, etc.)

NASA utilizes multiple parametric models that work well for "normal" scientific satellites with "normal" instruments where 50 to 70% of the estimating work is done without penetrating to lower levels of the project's work-breakdown structure (WBS). Projects viewed as not 'normal' may require more specific estimates, but high level parametric estimates provide serviceable data for this early stage of the project lifecycle.[2]

A joint cost-schedule confidence level (JCL) represents the probability that the project's cost will be equal to or less than the targeted cost and schedule will be equal to or less than the targeted schedule date. To calculate a JCL, the project uses a process that combines its cost, schedule and risk into a single model that can generate a probabilistic assessment of the level of confidence of achieving a specific cost-schedule goal. For high level projects of large investment thresholds a JCL is conducted in support of KDP-C[3]. The rationale is for two primary reasons: 1) the goal of the lifecycle review is to assess the project's detailed plans and the project's plan is well defined, and 2) this is the timeframe in which NASA is committing to external stakeholders. The Agency uses this assessment when considering its external commitment (KDP-C) as one means of ensuring the project has a robust plan with costs linked to schedule, where both are informed by risks.

Once a baseline is approved, NASA policy does not require a project to maintain the artifacts used to calculate the JCL. However, the Agency does utilize a variety of performance metrics to assess how well the project is performing against its plan. If these metrics show that a project's performance varies significantly from its plan, the project may need to re-plan, but Agency policy only requires a repeat

---

[1] Key Decision Point (KDP) – B. In accordance with NASA's NPR 7120.5E this milestone correlates with a Systems Readiness Review in most cases.

[2] Hamaker, Joseph; '*Do More Details Imply Better Accuracy in Cost Estimates*?'; Parametric World, ISPA, Summer 2010 Vol. 29 No. 3

[3] Key Decision Point (KDP) – C. This milestone can be correlated with a Preliminary Design Review.

calculation of the JCL in the event the project requires a rebaseline. JCL analysis can provide valuable insights as a management tool; however, the only Agency requirement for JCL is at KDP-C.

Parametric analysis becomes less fruitful to the project manager (PM) as a project baseline plan is properly defined. Typically, PMs deal with project management artifacts such as the integrated master schedule (IMS), cost and budget figures supported by low level contracts and/or subject matter experts (SME) input, and a risk posture defined by a detailed master risk list and historical heuristics on reserve posture. Parametric analyses, while justifiable due to their basis in historical data, rarely tie well to the project management artifacts. This has resulted in a culture where cost/schedule estimators (parametric analysts) are put at odds against project managers due to their historical and inconsistent inability to link their analyses in a frame work where they can be communicated and presented in an "apples-to-apples" comparison. The primary shortcomings to this cost estimating approach (in comparison to a tool based on program artifacts such as JCL is twofold: it does not reflect key engineering design parameters, and it does not reflect key management decisions[4].

The issue of not being able to effectively link parametric analysis (done at KDP-B) to cost management build up estimating (done at KDP-C) has resulted in inefficiencies in the broader NASA programmatic community effectiveness. In addition, cost estimators at NASA have observed that the Joint Cost-Schedule Confidence Level (JCL) analyses required by NASA policy to occur at PDR/KDP-B, have yielded suspiciously narrow Coefficients of Variation in JCL cost S-curves. This gives rise to concerns within the cost community that projects, overly reliant on their own SMEs to provide uncertainty ranges, are missing opportunities to incorporate significant uncertainties into their estimates.

NASA's Cost Analysis Division (CAD), Booz Allen Hamilton and PRICE Systems collaborated to conduct research into linking parametric estimates to programmatic artifacts in a manner that would elevate parametric estimates and allow Programs and Projects to apply the historical lessons that make parametric estimates so powerful and accurate. This research brought together an 'LPEPM Team' of parametric and programmatic cost estimators, model developers, software developers, schedulers, risk analysts and practitioners ranging from junior analysts to Ph.D. thought-leaders to think through and articulate a process by which parametric cost estimates could be linked to programmatic artifacts in a manner that takes maximum advantage of the best each has to offer. Specifically, the collaborative research evaluated the feasibility of a parametric cost model "informing" a JCL model and vice-versa via iterative methodology.

The goal of the research is to demonstrate that it is possible to mitigate the aforementioned inefficiency and thus improve the valuable characteristics of both methodologies. The benefits of creating a more synergistic relationship between parametric analysis and the project management artifacts are plentiful, examples include:

1. Traceability between parametric estimates to project management artifacts (schedule, budget, risk) would encourage better communication at KDP-B between the parametric analysis community and the project management community.
2. Continuous linkage between parametric and project management artifacts would allow projects the ability to, quickly and defended by data driven metrics, revisit their initial estimates based on technical and programmatic changes. For example, this could increase the

---

[4] Prince, Andy; '*Why NASA's Management Doesn't believe the Cost Estimate*'; Engineering Management Journal, American Society for Engineering Management, March 2002 Vol. 14 No. 1.

accuracy, precision, and credibility of NASA's continually fluctuating budget environment as well as provide impact statements to definable technical changes.

3. Ability to provide calibration inputs, informed by the project's performance data, back into the parametric analysis (models). This would increase the usefulness and timeliness of the parametric analysis for future projects.
4. Ability for parametric "outputs" to be used as "inputs" to JCL tools, thus augmenting, and reducing the reliance of, subjective inputs by SMEs.
5. Finally, all four of the above mentioned benefits could be utilized by the Agency's independent entities to enhance NASA's ability to quickly and accurately assess the realism of a projects plan – regardless of where it is in the lifecycle.

LPEPM Methodology

Initially, the LPEPM Team observed that there are two distinct scenarios for linking parametric and programmatic approaches, *depending upon when in the project lifecycle the attempt to link occurs*. These constitute two separate scenarios with processes unique to each. Scenario #1 introduces parametric estimates mid-lifecycle, after project artifacts (i.e., schedule, risk list, cost estimate/budget, etc.) have been developed. It focuses on adapting the parametric model to programmatic products as a means of building credibility in the eyes of the PM. It is the most likely scenario to be encountered. Ultimately, the preferred approach that promotes best practice is Scenario #2 where parametric and programmatic are linked from the beginning. Artifacts are developed in tandem, cross-informing one another from the outset. This paper makes recommendations regarding achieving the conditions for enabling Scenario#2. Given the LPEPM's Team's working example of an ongoing large-scale NASA project (hereafter referred to as 'the Project') already in progress, Scenario #1 was the primary focus of this research.

In this most likely scenario where a parametric estimate is developed independently for a project which already has mature programmatic artifacts, the driving issue is building credibility for the parametric estimate in the eyes of the PM. The LPEPM Team defined an iterative, step-by-step process for aligning the two, where the iterative process itself builds confidence in the parametric output. Parametric cost estimates are often used early in a project lifecycle as required documentation and to establish a budget wedge, but fall to disuse as project-specific artifacts render parametric OBE. The LPEPM Team recognized that it is incumbent upon the cost estimator to build the credibility of the parametric model in the eyes of the PM. The LPEPM Team defined a four-step methodology, visualized via <u>three</u> custom dashboard iterations, for alignment of parametric to programmatic:

1. **COLLECT DATA** to compare unmodified parametric and programmatic estimates, side-by-side;; align costs by mapping the parametric model Product Breakdown Structure (PBS) to the project's WBS;
2. **NORMALIZE** technical scope and key assumptions between parametric and Project artifacts to ensure estimates cover the same content; un-constrain the parametric schedule to create a "should cost" estimate;
3. **ANALYZE** discrepancies between the two estimates to determine if divergences indicate potential areas of hidden cost risk or programmatic blind spots;
4. **OPTIMIZE** by adjusting both models, each informing the other, with the parametric schedule constrained to reflect the IMS and the JCL reflect of the parametric cost, thus allowing for a clean "will cost" comparison of two models.

Each step is intended to enhance credibility and acceptance of the parametric model, while extracting the best qualities of each to inform the other. As stated above, the LPEPM Team created a **Dashboard** tool to aid in visualizing side-by-side comparison of parametric and programmatic products. At each of the three iterations, the dashboard shows (a.) predictive cost estimates, (b.) predictive schedule estimates (c.) costs phased by year, and (d.) comparative cost drivers. Furthermore, to expedite production of the Dashboards, the LPEPM team developed a companion **Data Template** to standardize collection of the data needed for export (from models) and import (to Dashboard) in a manner agnostic of tool or model used.

STEP #1

This first step was designed to capture programmatic and parametric outputs as they exist in their rawest form, regardless of commonality of scope or degree of calibration, and to lay those outputs alongside one another in a common format for accessible, side-by-side comparison. It serves as the starting point for initiating subsequent conversations about scope mapping and calibration. Harvesting data for import into Data Templates will initiate the estimator to the process and complexities associated with finding and exporting proper, relevant data sets. This first step serves as a visual reference against which to compare successive dashboard iterations.

In order to collect data to make raw comparisons, this process was taken (a.) obtain the parametric and JCL models, (b.) map parametric PBS to Project WBS using schedule Unique Identification Descriptors (UIDs) to apportion costs and (c.) produce Dashboard Iteration #1. The result was raw estimates with costs cross-mapped to provide a Baseline comparison of the two models. The methodology is outlined as follows: The LPEPM Team first captured and juxtaposed the Project parametric and program (JCL) estimates to the raw parametric estimate (from TruePlanning™) for direct comparison. As mentioned above, the specific parametric example was from an initial large-scale NASA project currently underway. Its point estimate was collected and exported to the Data Template. This predictive cost estimate also supplied S-curve data points. Likewise collected was the most-likely (i.e., deterministic, not stochastic) schedule output generated in TruePlanning™. Lastly collected were the parametric cost/ budget phasing data by year, as well cost driver data points (to produce or replicate any Tornado Chart outputs). In parallel, the LPEPM Team collected the example project's current JCL from the JCL tool (Polaris™) used by program management and exported the corresponding four data sets. If no JCL had existed, the Team would have used the ongoing project's existing probabilistic cost estimate, Integrated Master Schedule (IMS), phasing plan, and risk list in lieu of Polaris™ JCL outputs.

At this point, the LPEPM Team mapped costs between the parametric and programmatic models using IMS/JCL UIDs. In all, five UIDs were targeted successfully: 6008 - Management; 6016 - Design; 6084 – Test & Evaluation; 2494 - SE&IT; 2728 – Manufacturing & Assembly; and (not shown) 2728 – Avionics (software).

Figure 1 illustrates the mapping. The colors in figure 1 represent the mappings between models, so that corresponding parametric activity costs were apportioned to UID activities, using the mapping percentages below. Of note is that since the output into the JCL tool (Polaris™) is an S-curve for each UID category, a **two-part** technique was constructed. First, assuming nominal uncertainty ranges on the cost drivers, Monte Carlo simulations were conducted using TruePlanning's™ companion application to Oracle's Crystal Ball™, for each parametric output task-activity (i.e., the source columns below). Second, for each trial of these simulations, the outputs were shredded to the UID activity categories (i.e., the target rows below), using logic from the TrueMapper™ companion application, to build up S-curves/tables for input into the Polaris™ tool for JCL. The new process successfully generated simulations for cost-risk that informed the JCL model for five UIDs.

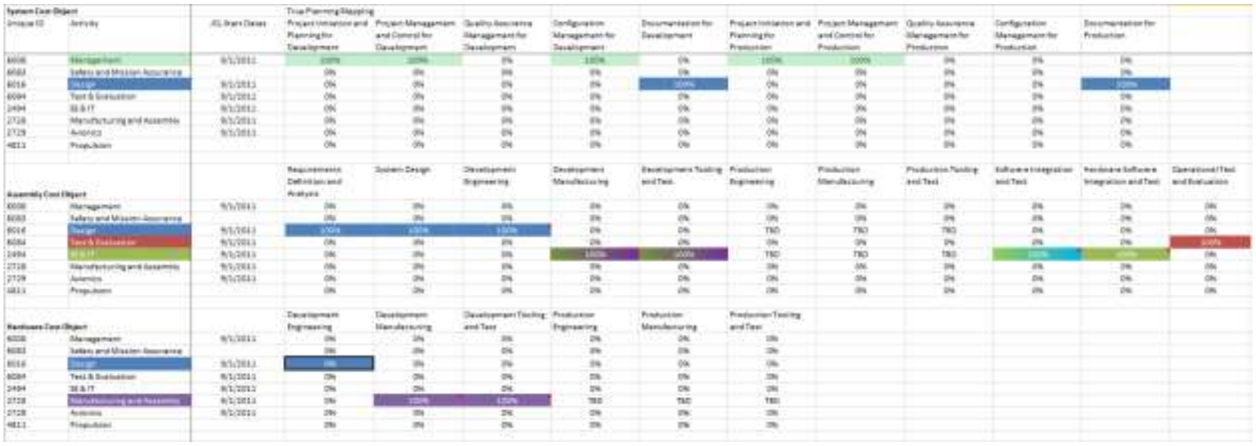
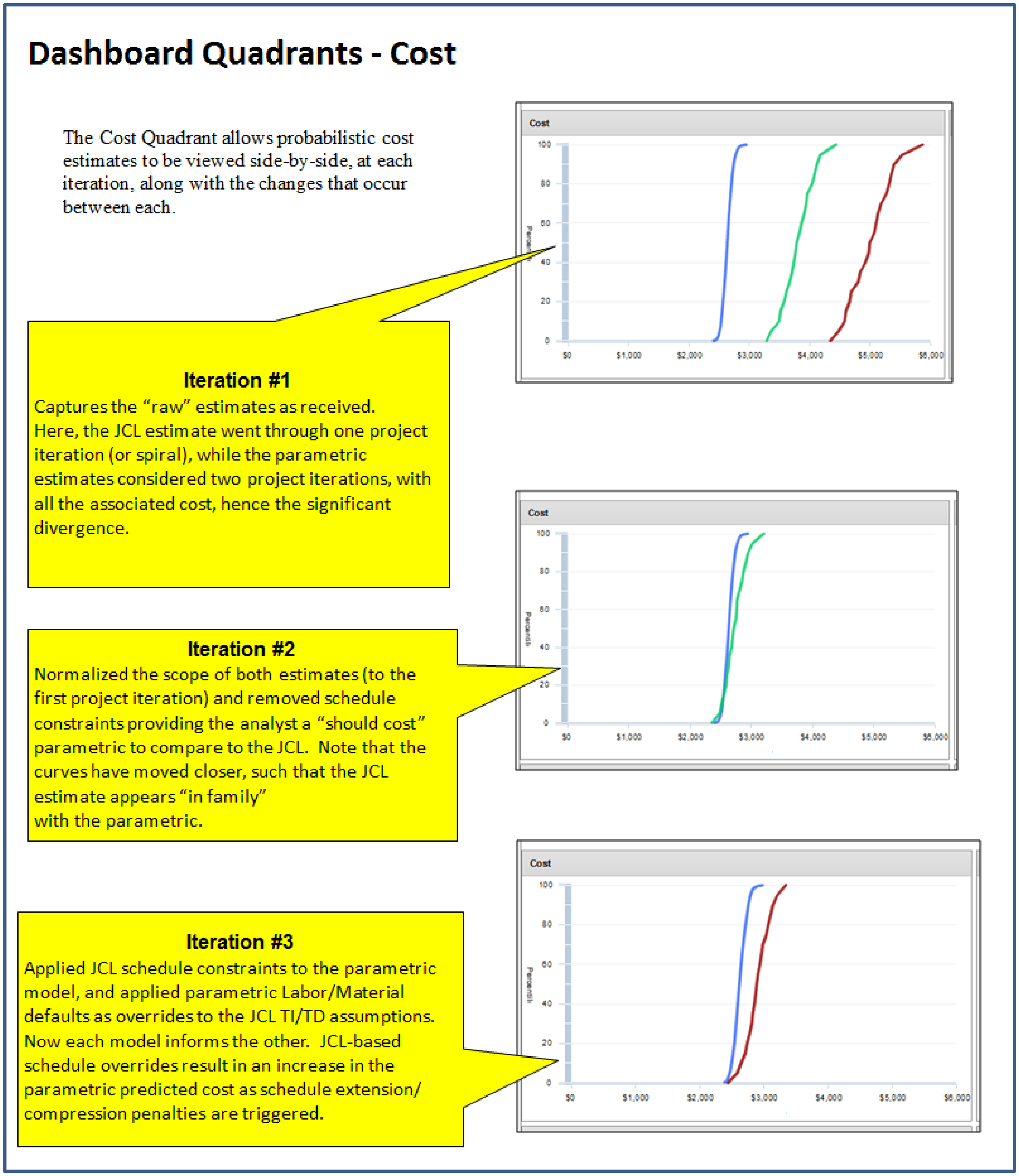


Figure 1

Finally, the templates were imported into the Dashboard Tool to produce Iteration #1. Initial observations from this first step were reasonable. First, parametric cost models may rely on 3rd party intermediary applications (Crystal Ball™, @Risk™) to produce probabilistic outputs. Second, schedule data from the parametric tool should be incorporated as-is, even if not probabilistic. Third, if the initial parametric schedule is constrained, the constraints should be left in place. If it is unconstrained, no changes should be made. (In this case, the example parametric model had been constrained, but not in the manner the LPEPM team re-applied in Iteration #2). Fourth, if the Project's programmatic artifacts are not sufficiently mature to integrate in a JCL tool, populating the template for the programmatic side could be considerably more labor intensive. Finally, there is no need to worry if the parametric schedule is deterministic. It will show as a vertical line on the schedule Dashboard, as was the case here. Figure 2 shows the Dashboard cost display for all three iterations.

STEP #2

The second step provides a scope-normalized "apples-to-apples" comparison of parametric to programmatic, in which the parametric cost model is adjusted to reflect content in the way that the Project does. This step builds credibility by starting to address the "that's not my project" objection. Removal of schedule constraints in the parametric tool results in a cost-optimized most likely schedule – based on logic contained in the tool. While an abstraction, it provides a baseline against which to gauge the effect of compression or extension penalties incurred when project-based schedule constraints are applied in Iteration #3. Re-application of the two-part process constructed above (yielding S-curves by UID activity from parametric risk simulation) provides a de-facto independent crosscheck of the Project IMS and technical scope coverage.



Dashboard Quadrants - Cost

The Cost Quadrant allows probabilistic cost estimates to be viewed side-by-side, at each iteration, along with the changes that occur between each.

Iteration #1
Captures the "raw" estimates as received. Here, the JCL estimate went through one project iteration (or spiral), while the parametric estimates considered two project iterations, with all the associated cost, hence the significant divergence.

Iteration #2
Normalized the scope of both estimates (to the first project iteration) and removed schedule constraints providing the analyst a "should cost" parametric to compare to the JCL. Note that the curves have moved closer, such that the JCL estimate appears "in family" with the parametric.

Iteration #3
Applied JCL schedule constraints to the parametric model, and applied parametric Labor/Material defaults as overrides to the JCL TI/TD assumptions. Now each model informs the other. JCL-based schedule overrides result in an increase in the parametric predicted cost as schedule extension/ compression penalties are triggered.

Figure 2

In order to normalize scope, this process was taken: (a.) refine mapping of parametric to Project to normalize scope and assumptions, (b.) un-constrain the parametric schedule, and then (c.) produce Dashboard Iteration #2. The result was scope-normalized models allowing for "apples to apples" comparison of Project estimates against "should cost" estimate. The LPEPM team recognized that a common complaint of PMs about parametric estimates is, per above, that "they don't reflect my

Project." In this second step, the LPEPM Team sought to normalize content scope and assumptions so that the parametric estimate covered the same content as appeared in the Project IMS and estimates. The methodology is outlined as follows: The LPEPM Team revisited the mapping of parametric PBS to the Project WBS performed in Iteration #1, this time to ensure that scope aligned completely. The Team identified holes, overlaps and conflicts of scope and adjudicated with JCL analysts. The intent was to get the parametric model to reflect scope as the Project captured it. Schedule constraints were removed from the parametric schedule to generate a cost-optimized "should cost" parametric estimate. No change to Project inputs were made in Iteration #2. Most likely parametric cost modeling was re-run. The two-part construct was again executed to yield UID S-curves. The Team then re-ran the Dashboard as Iteration #2.

Observations from this second step were straightforward. The parametric tool used was organized around a PBS based on a MEL-driven weight statement, while Project artifacts are typically based on a WBS, usually driven by the IMS. Mapping between the two models in order to get the parametric model to reflect scope recognizable by the Project is a critical step towards earning credibility. Defining the mapping was dense, tedious and time consuming. The LPEPM Team had individuals from the parametric modeling side as well as from the Program's JCL team cross-walk the two models. Some activities align one-to-one (management) but others are more obscure (e.g. design). No change was made to any of the programmatic elements for Iteration #2. The goal here in this step is to bring the parametric into line with the programmatic. Removing constraints from the parametric schedule provides a cost-optimized schedule as a baseline reference. The only constraint the LPEPM Team recommends is a credible Project start date so that phasing aligns.

STEP #3

The third step does not produce another iteration for Scenario #1. It however investigates discrepancies in predictive cost, schedule, phasing profiles and cost drivers. With paired data sets presented visually alongside one another after two distinct iterations, the Dashboard enables accessible observation and analysis. Resulting from this third step is providing the PM a credible tool for cross-checking programmatic artifacts. Divergences between the two models could be indicative of areas where potential cost risk resides or programmatic blind spots exist. The methodology is outlined as follows: For cost discrepancies, observe if the JCL and parametric cost S-curves intersect. If they do not, the implication is that the project's execution plan falls outside the historical record captured in parametric CERs (i.e., a reality check). Ask next what discrepancies exist at the sub-element level and what accounts for them. For Schedule issues, evaluate the parametric model for the effect on cost of un-constraining the schedule between Iterations #1 and #2; similarly, compare how close is the Project schedule to Iteration #2's unconstrained parametric schedule. {Compare the latter's costs and schedules to those of Iteration #3 after re-constraining the schedule within Step #4.} Likewise in Iterations #2 and #3, ask if the model invoked compression or extension penalties, and is it doing so appropriately. Finally, check if there are activities in the IMS whose duration diverges significantly from the "cost optimized" parametric schedule. For phasing discrepancies, ask if the curves look similar. Check if each year falls within the expected budget for that year. Regarding cost drivers and apparent discrepancies, parametric models offer different visibility to inputs. For example, the TruePlanning™ model allows for sensitivity analysis within the tool, as well as robustly through Excel™-based companion application. It could interest a PM to observe the effect of weight or mission complexity or organizational maturity on the overall cost model.

STEP #4

The fourth step was intended to achieve two ends: (i.) inform the parametric model with enough project-specific calibration to earn acceptance as a credible and valid crosscheck instrument, and (ii.) allow an opportunity for each model to inform the other. This fourth step produces Dashboard Iteration #3, which is the first juxtaposition of programmatic and parametric estimates, wherein the latter has been scope normalized and reasonably adjusted to reflect key realities of the Project, including schedule constraints informed by the project's schedule. The opportunity to cross-calibrate is the means by which to allow the best elements of one model to be incorporated into the other. Iteration #3 provides an opportunity for the programmatic results to be re-run using inputs from the parametric model (if desired) to override existing inputs

In order to optimize by way of cross-informing both models, this process was taken:
(a.) constrain parametric schedule per IMS, (b.) apply parametric outputs to JCL (e.g. TI/TD; uncertainty) as appropriate and (c.) produce Dashboard Iteration #3. The Result is both models informing one another using the best qualities of each, providing a "will cost" comparison to Project. The methodology is outlined as follows: The parametric schedule was constrained with dates from the project's deterministic (not probabilistic) IMS. Project start and end dates were captured and populated because the parametric model looks for Start/End dates of activity phases like design, manufacturing, I&T, etc. {Lesson: Less is better; err on the side of fewer schedule constraints than more.} The intent of constraining these dates is to trigger schedule compression/ extension penalties contained within the model. Next, key parametric uncertainty inputs were adjusted based on information available from the Project. For example, consider examining weight (mass) of structure vs. electronics (where available), corresponding complexities for manufacturing, engineering and functional complexities, organizational productivity and software size. Again, as in Step #2, most likely parametric cost modeling was re-run. The two-part construct was then executed to yield UID S-curves. Re-informed Project inputs/ models were based on new parametric inputs. An example includes Labor/Material assumptions applied as overrides to JCL time-dependent/time-independent (TD/TI) assumptions. The Team then re-ran the Dashboard as Iteration #3.

Observations from this fourth step were as follows. Populating the Data Templates for each of the three iterations were time consuming and should be factored into the analyst's production schedule. Our parametric tool uses a 3rd party application to run Monte Carlo simulations. Depending on the size of the models or the number of analysts with access to the tool, this constraint can become a logjam. When aligning the parametric schedule, a few key start/end dates work fine; but don't over-constrain. Remember: the intent of this activity is to produce a credible independent crosscheck of Project artifacts as a means of (a.) flagging potential blind spots, (b.) providing PMs and RMs with a different perspective on risk and cost drivers, and then (c.) providing a credible crosscheck of Project artifacts, findings, estimates and projections based on historical actuals captured in CERs. It is **not** the goal to have the two models arrive at the same estimate. Discrepancies between the two become potential areas for investigation. The process is intentionally iterative, with the Dashboard specifically designed to capture each iteration for immediately accessible reference: All this effort is done as a means of earning

Overall Observations

The Dashboard and Data Collection templates are designed to be tool-agnostic artifacts, such that choice of parametric tool, JCL tool, scheduling tool, enabling applications should have no effect. Almost all tools export to MS Excel™ or to XML. The template is designed to accommodate any of these outputs, while providing clear understanding of the data sets required. The Team suspects there is value still to be extracted from a parametric tool – particularly vis-à-vis uncertainty. There is value to getting parametric estimators and models brought back to the table, but the meaningful historical lessons are "locked in the ice" of the CERs themselves. This bears further effort and discussion. There were interesting observations enabled by the Dashboard. At Iterations #2 and #3, we noticed the 1st quadrant comparing the two cost S-curves and considered the significance. One should expect the JCL (i.e. programmatic) probabilistic cost curve to intersect the parametric curve somewhere. The parametric curve represents all of history via its CERs. If the project's probabilistic cost curve does not intersect somewhere, it suggests that the project's estimate is inconsistent with all like projects and programs that have gone before it. This could be due to inputs to the parametric model that need to be revisited. Or it could be that the Project estimate is unusually optimistic (or pessimistic).

There was little that could be concluded concerning suspiciously narrow Coefficients of Variation in JCL cost S-curves. This issue may be an item for further research.

Lessons Learned

The act of mapping the TruePlanning™ PBS to the project's JCL was instructive, and all members of the LPEPM Team regarded the activity as extremely valuable. Mapping was tedious and difficult, with some cost elements mapping neatly (Program Management) while others far more oblique. It is difficult to overstate how critical this activity is to the downstream process, so having the right people in the room with significant understanding of the nature of the mapped artifacts as the effort is commenced is of upmost importance.

The mapping activity forced interesting conversations. For instance, the parametric model included a cost object with an undecipherable name. The LPEPM Team scoured the JCL IMS and could find nothing corresponding. We engaged the JCL analyst from the Project who was able to help map this quickly for us.

Populating the template proved more nettlesome than was expected. The LPEPM Team initially assumed that items would export easily from JCL and parametric tools. Generally, Polaris™ exported cleanly (full disclosure: the team coding the dashboard is the same that coded Polaris™, so there was a transitional advantage). We encountered unexpected issues exporting similar outputs from TruePlanning™. In particular, extracting data through 3rd party tools (i.e. Crystal Ball™) was more difficult than anticipated. Also, 3rd party Monte Carlo tools can be slow.

The study was chartered originally as "research" only; our developer was quite correct in offering that a more robust, elegant solution was completely feasible with more time. Executing "one off" solutions of each of the UIDs, for each of the Iterations, was not optimal

The population of the Template for Dashboard Iteration #1 should be used as a chance to assess if and where logjams may come into the mapping effort. How many analysts have access to the tool suites used? Does one analyst have the parametric tool, and somebody else have the JCL tool? How long does it take to run simulations to generate S-curves? How many of your analysts know how to perform Monte Carlo modeling and export to the Template? These are all elements we encountered as minor roadblocks.

Conclusions

LPEPM as a concept appears to work. The Team used actual JCL and parametric models -- developed independently – in a real-world test case to demonstrate the process, and to refine supporting visualization tools. The value of the LPEPM process is an outgrowth of its iterative nature. Each step builds upon the last, provides a unique crosscheck, and builds confidence in the credibility of the parametric model.

1. Iteration #1 mapping cross-walks costs allowing comparison of cost elements between models
2. Iteration #2 normalizes content through a mapping process that also acts as a crosscheck of the Project WBS for omissions or overlaps and produces a "should cost" comparative estimate
3. Iteration #3 informs the parametric model using Project inputs enabling "will cost" comparison

Two distinct LPEPM scenarios exist – depending on when in the lifecycle linkage occurs – each with its own unique implementation process.

1. Scenario #1 (most likely scenario) envisions adapting a parametric estimate to a mature Project with programmatic artifacts that already exist, including an IMS, risk list, and cost estimate or budget, and possibly a JCL
2. Scenario #2 (preferred scenario) assumes LPEPM implemented from Project inception, with parametric products informing the development of programmatic artifacts from the beginning

Parametric models can be informed using programmatic inputs, and vice versa. Linking parametrics to programmatics fills several long-standing shortcomings in both analyses and is something both communities should strive to implement in their programs. This allows an 'apples-to-apples' comparison between the program and parametric estimates by ensuring consistent structure, technical scope, and program execution assumptions. It allows project manager decisions to inform the parametric model through the use of project-centric independent variables and the calculation of extension/compression penalties. In some cases allows parametric estimates to directly inform JCL components. It may inform DoD's should-cost/will-cost analysis. Finally, it extends the value of parametric estimates into the project execution phases.

Future areas of research and investment will continue to strengthen the link between these disciplines. These include identifying additional programmatic attributes in our CERs and SERs, finding ways to include schedule uncertainty in our compression/extension penalties, and developing parametric estimating tools to better interface with programmatic artifacts