Schedule Uncertainty Quantification for JCL Analysis

Sally Whitley
August 13th, 2014

JOHNS HOPKINS UNIVERSITY
Applied Physics Laboratory

Motivation

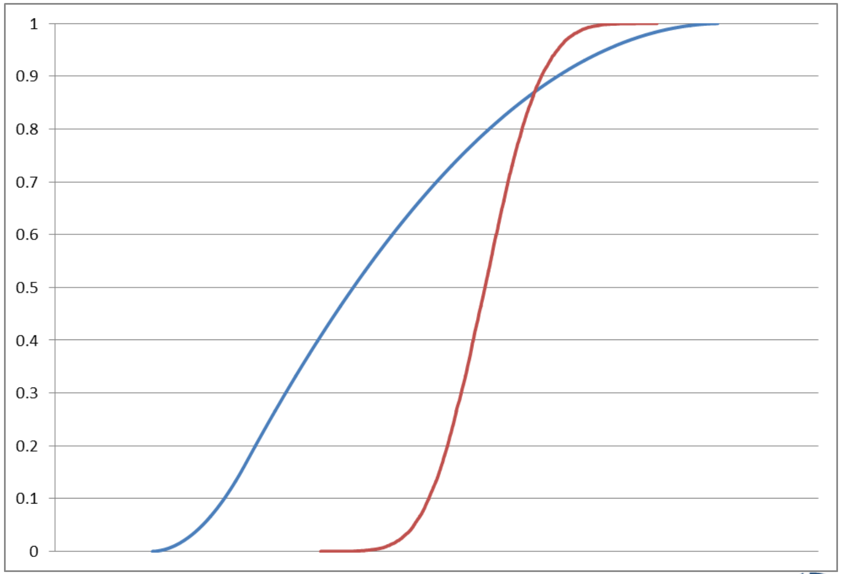
- **The schedule portion of JCL analysis**
  - Causal factor for costs
  - Driving issue for JCL results
  - But how do we tackle it?
- **Historical data?**
  - At what level of detail should we be analyzing?
    - How does an actual schedule's behavior translate into an analysis schedule's predicted behavior?
    - What factors are good predictors for task level variability?
  - (At what level of detail do we even have data to look at?)
    - We need a PDR schedule and a launch schedule
    - Tasks between the two schedules have to line up to get a valid comparison
  - How does task behavior translate into summary task behavior? How does task behavior translate into mission-level schedule behavior?
    - Ultimately, we want to know the risk to the launch date
    - We analyze at the lower level to gain insight into
      - How schedule topology affects launch readiness outcomes
      - What tasks may threaten the critical path (where are pockets of reserve inadequate)

Problems for practitioners

- **Guidelines for assigning schedule uncertainty at the task level in analysis schedules**
  - Often generated from mission-level schedule growth data
  - Stratified into qualitative low-med-high uncertainty categories (to be assigned by the JCL analyst)
  - Missing guidelines for prediction approach
- **Organizational data are often too dirty or too sparse to facilitate an internal analysis**
  - We had a viable data set from one mission
  - Cleaning it up into an information-rich format was painful
  - We had just over 1200 records to work with, which were task level data from an actual mission schedule – different from task level data from an analysis schedule!
  - Summary task level behavior could be gleaned from the data as well

What's wrong with the mission level schedule growth data?

- **Mission level schedule behavior does not predict task level schedule behavior**
  - The JCL is intended to work *the other way around* – We are trying to use task level information to analyze mission level outcomes
  - Even in a simplified case, where a single, dependent series of successive tasks leads up to the mission launch date, we can't apply mission level data to task behavior
    - Outcomes will be biased high
    - The problem becomes greater with more complex schedule topology



The blue line shows the "true" distribution of launch readiness dates.

The red line shows the distribution of predicted launch readiness dates when the mission distribution is applied at the task level.

In this example, expected schedule is overestimated, while overall schedule uncertainty is underestimated.

What's wrong with low-med-high?

- **No rigorous approach to assigning these classifications**
  - Predictor variables should be objective and quantifiable
  - In essence, this approach means we are assigning the outcome we already decided we should see in our analysis output. What's the point?
- **Concerns about double-counting**
  - Uncertainty ranges should be independent of the project risk list, since risk effects on schedule should be quantified and applied separately
  - If a subsystem is subject to a large number of risks, an analyst could have a tendency to assign the subsystem an uncertainty range of "high", in addition to the risks already affecting it. This may or may not be appropriate.
- **Concerns about under-counting**
  - Historical data may support the use of a larger uncertainty range than an analyst would assign based on intuition.
  - Particularly, a task within a subsystem with few specific risks might seem "benign" and be assigned small uncertainty, when in reality the task may be subject to large uncertainty independent of specific risk

How can uncertainty be quantified?

- **Uncertainty is independent of risk!**
  - High risk subsystems can have low uncertainty, and the other way around
  - In order for this to work, JCLs need to have complete, thorough risk lists
  - Uncertainty needs to capture the unidentified risks – how?
    - Historically manifested risks are a good starting point:
      - It is not valid to try to isolate historically manifested risks out of the dataset
      - That would be to assume that the project risk list is exhaustive
      - Is that even possible…?
    - Historically manifested risks represent
      - Unanticipated risks that had a negative effect on project outcome
      - Identified risks that were not successfully mitigated
  - It is likely that the historical data can only provide a subset of possible outcomes
    - Appropriate to assume that the population distribution has a tail
    - The history can provide insight into the expected outcome, along with an idea of what the standard deviation should be

Uncertainty distributions can be defined using parameters from historical data, as long as

1. Manifested risks are **not** removed (to account for unidentified risks)
2. The distribution's "tail" is allowed to **grow past the historical data** (to account for the fact that the history is only a subset of possible outcomes)

But how do we get to this data?

- **There are two different ways to think about schedule uncertainty**
  - Uncertainty measured as percentage growth (+/-) from original estimate (standard approach)
  - Uncertainty measured as absolute days delta from original estimate (not often used currently, but may warrant further research)
- **For the percentage growth approach, we need a PDR schedule and a launch schedule from the same project**
  - Tasks have to be lined up and compared
  - Mapping issues will cause a lot of data loss
- **Data for schedule ranges should be analyzed at roughly the same level of detail as the data to which it will be applied**
  - Task level data applies to task level schedules
  - Summary level to summary level, etc.
  - For analysis schedules, we need to use historical data at a higher level of detail than what's found at the task level

A word about percentage growth vs. absolute days

- **The current approach uses deltas between PDR schedules and launch schedules to predict a distribution of the growth of a task's duration**
  - This assumes that a CAM's estimate of the duration of a task will not increase in accuracy from experience with one program to the next
  - Danger: If the CAM's experience with a historical program does inform his/her estimates for task durations of future projects, this approach will double-count schedule growth
- **What if, instead, we use the absolute durations of different types of tasks to predict corresponding tasks?**
  - The approach is robust to CAMs' different levels of learning and experience
  - Data needs are reduced
    - No PDR schedule is required
    - No task-mapping is required
  - But finding the distributions of the durations in absolute days of different types of tasks presents a new challenge
- **Worthy of further research**

Challenges with the data analysis

- **When we tried it…**
  - Task mapping was arduous
  - There was a lot of data loss
- **Predictor variables were difficult to categorize**
  - Subsystem? Type of task? Milestone?
  - We settled on subsystem because it was a better predictor than the other two variables (but it still wasn't always great, depending on the subsystem)
- **After we fit distributions to the data, more problems arose**
  - Uncertainty ranges were huge at the task level, not reflected in subsystem uncertainty
  - Applying these ranges to a JCL model schedule resulted in absurd results
    - Launch readiness date schedule growth over 80 months
    - Recursively applying the uncertainty ranges to their own dataset for a test resulted in the same absurd results

What could cause this?

The explanation

- **Tasks on or close to the critical path behave differently from tasks not on or close to the critical path**
  - Critical path tasks may benefit from greater resources
  - Schedule pressure may play a role in how efficiently people work
  - Non-critical path tasks, not under schedule pressure, may grow significantly without becoming a threat to the project
- **In a project schedule, there are far more non-critical path tasks than there are critical path tasks**
  - Non-critical path tasks bias the results of the analysis by overestimating uncertainty in critical path tasks
  - An analysis schedule has a higher percentage of critical path tasks than a detailed schedule – the bias is exaggerated further
  - There is no (efficient) way to address this problem in a JCL, which necessarily has an unknown critical path

Now what?

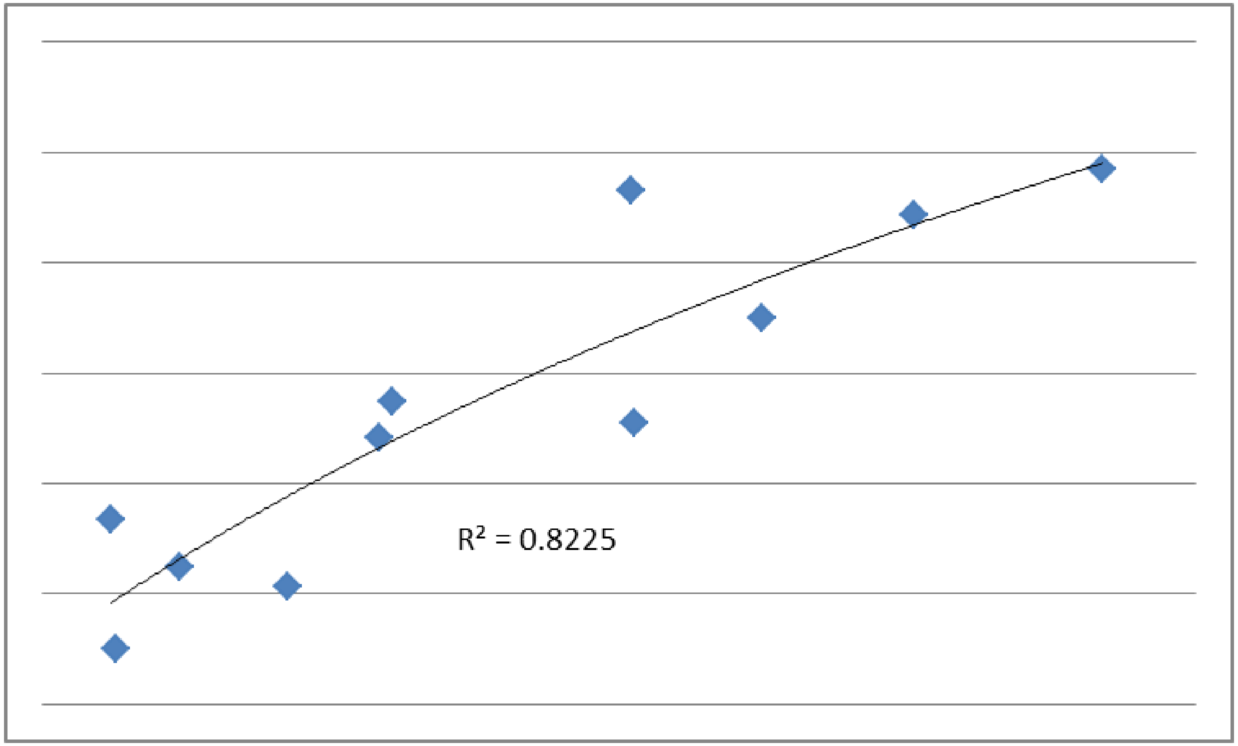
All was not lost!

- **The task level analysis wasn't for naught**
  - We learned that we could use subsystem behavior from historical missions to predict corresponding subsystem behavior in planned missions
  - We observed that lognormal distributions fit the data well
    - Produced reasonably good fit statistics
    - Generated distributions with appealing characteristics
      - Right skewed
      - Left bounded
      - Infinite right tail
      - Easily defined with two parameters, which can be data-driven
    - Use of lognormal for schedule uncertainty is well-represented in literature
  - This gave us the ability to use the shape we observed in the task-level data, whether the parameters associated with this data were valid or not (and they weren't)

So we had a distribution *shape*, but how could we translate what we observed at the summary task- and mission-levels into good parameters?

Going from summary level outcomes to task level inputs

- **We compared summary task-level PDR predictions to launch outcomes, with this breakthrough:**



Y axis = Actual duration

X axis = Predicted duration

Backing out the task behavior from the summary behavior

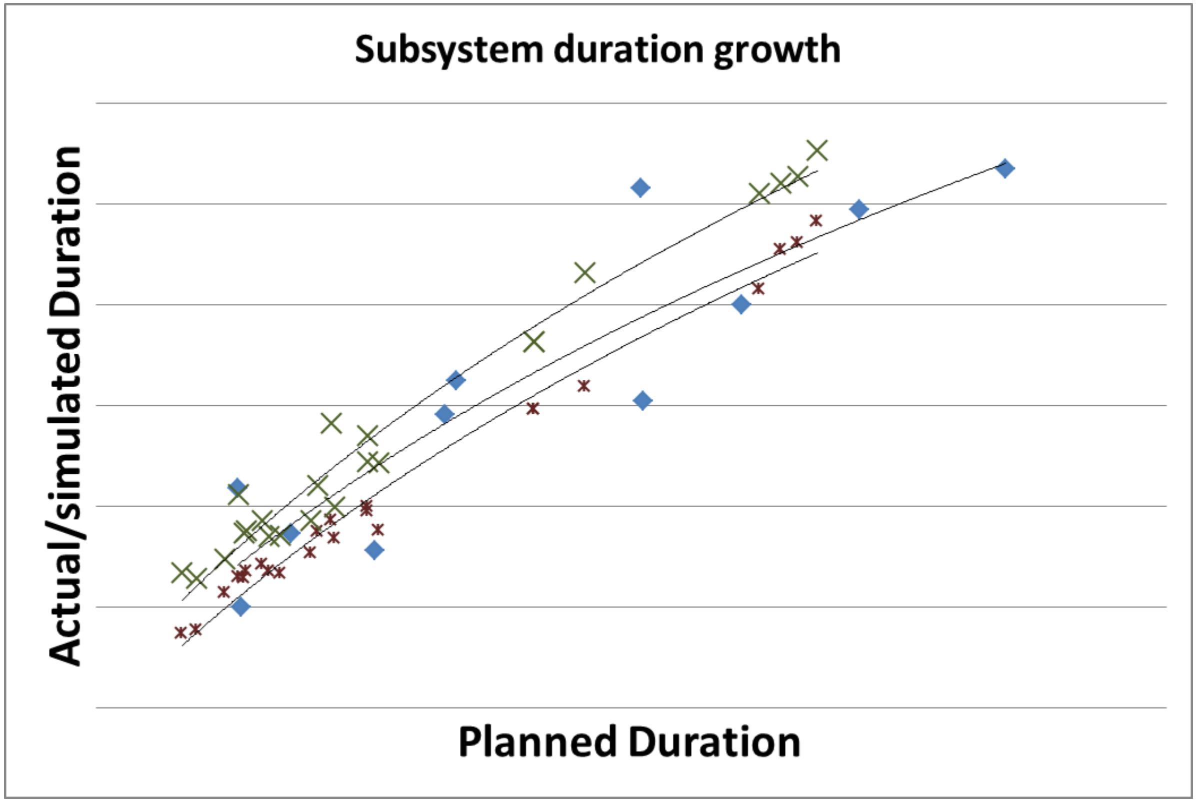
- **The graph above showed us that there was solid statistical evidence for predicting summary level behavior**
  - If applied correctly, we knew we could use the history as a cross-check for our analytical outcomes
  - But our JCL was at the task level – how could we back this out?
- **Some simplifying assumptions**
  - The distribution of the summary task is (roughly) the sum of the distributions of the tasks that were on the critical path (for any iteration of the JCL)
  - A sum of lognormal distributions is not defined, but it is often a close fit to other infinite, right-skewed distributions with fatter tails than lognormals (Weibull, Gamma, etc.)
  - If we assumed that all tasks within a subsystem were perfectly correlated with each other, we could solve to find the appropriate parameters for the tasks to achieve the subsystem outcome
  - Solution involved analysis of “task density”
    - Number of tasks within the subsystem
    - Average duration of tasks within the subsystem

Weaknesses

- **The approach isn't perfect**
  - We are back-calculating an expected result, at least at the task level
    - BUT! Schedule topology still drives (in some cases significant) differences between history and predicted outcomes, which is desirable
    - This problem could be solved without further ado if using an analysis schedule
  - Within a subsystem, there can be no variability of the critical path if all tasks are 100% correlated – except
    - Where specific risks affect individual tasks within the subsystem
    - The critical path between subsystems is dynamic!
  - Because we are concerned with maxima, correlation has a different effect on schedule uncertainty than what we are used to seeing in cost estimating
    - Higher correlation lowers expected value
    - The task density analysis actually was able to take this into account and correct for it at the summary level outcome

Testing the results

- The graph shows how predicted behavior at the summary task level compared to the historical summary task behavior



- Blue diamonds are historical outcomes
- Small red stars are predicted summary-level outcomes at the 50th percentile
- Green Xes are predicted summary-level outcomes at the 70th percentile

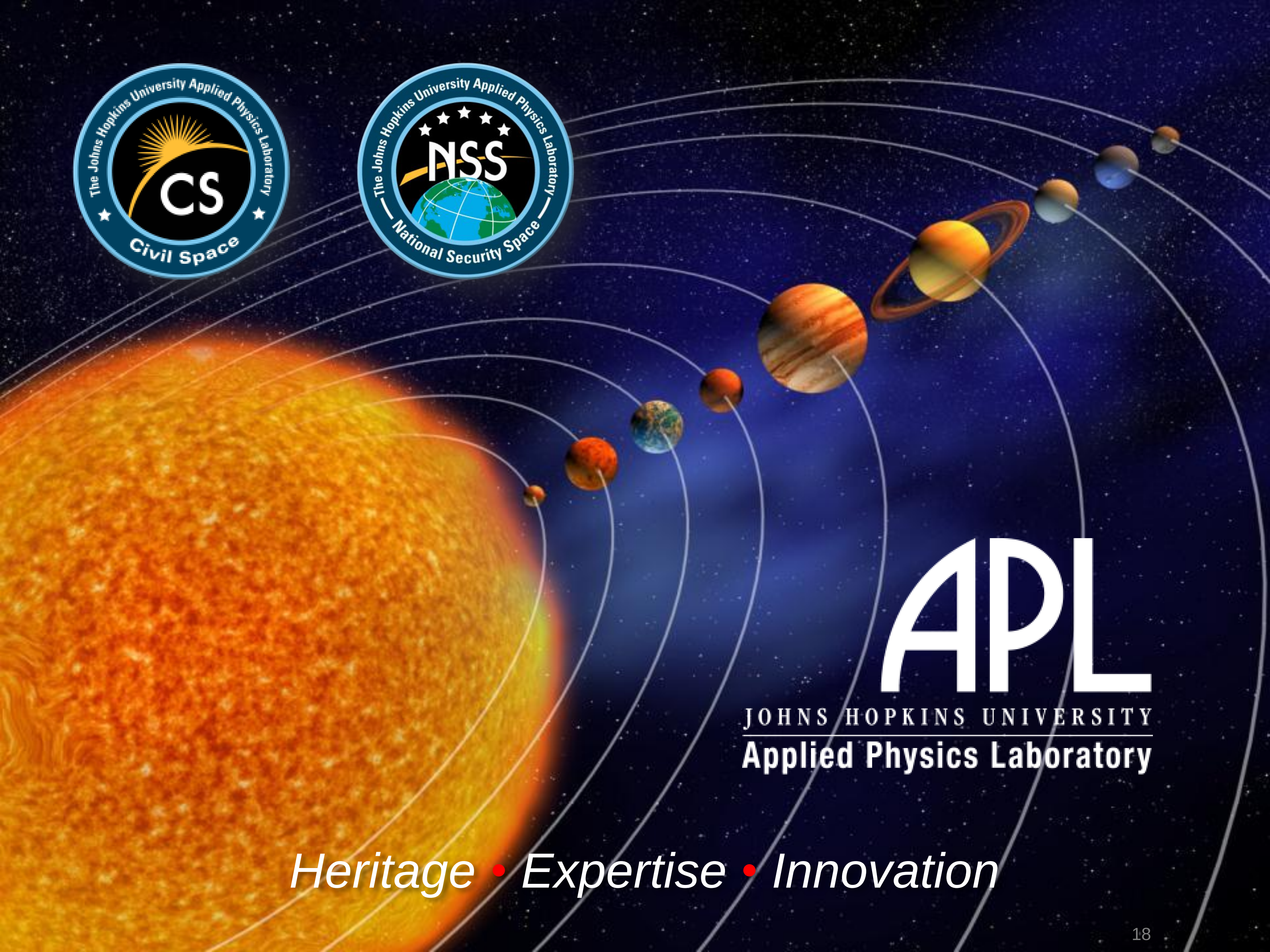
Summary of findings

- Guidelines based on mission-level outcomes using low-med-high uncertainty ratings are inappropriate for task-level behavior predictions
- Uncertainty ranges should be data-driven and assessed independently of specific risks
- Task-level data is problematic due to the effects of the critical path on task behavior
- Summary-level behavior can be used to back into task-level behavior by analyzing task density (imperfect)

Further study

- **Should the absolute duration of the historical tasks be used to predict the duration of the planned tasks?**
  - Implicitly takes learning and experience into account
  - Feasibility is a question mark – should be explored

- **Is there a way to solve for task-level behavior from summary task-level behavior assuming correlation within the summary task is less than 100%?**
  - Would answer static critical path concern
  - Would achieve analyzable results at the task-level


The Johns Hopkins University Applied Physics Laboratory
CS
Civil Space
The Johns Hopkins University Applied Physics Laboratory
NSS
National Security Space
APL
JOHNS HOPKINS UNIVERSITY
Applied Physics Laboratory
Heritage • Expertise • Innovation