An Overview of the NASA Software Cost Improvement Task:
How Does NASA Estimate Software Cost

Jairus Hihn
(Jet Propulsion Laboratory, California Institute of Technology)
Kevin Martin (Ames Research Center)
Lisa Van Der Aar (Glenn Research Center)
Manuel Maldonado (Goddard Space Flight Center)
Pat Benson (Marshal Space Flight Center)
Grant Tregre (Goddard Space Flight Center)
Karen Lum
(Jet Propulsion Laboratory, California Institute of Technology)

August 22, 2012

© 2012. All rights reserved.

Background and Purpose

Background

- The NASA Software Cost Improvement Task is a NASA wide task funded by the NASA Office of the Chief Engineer Software Working Group
- Team is multi-center and multi-disciplined

Purpose

- Improve software cost estimation across the agency
  1. Document current state of software cost estimation practices
     - Identify strengths and weaknesses
  2. Provide recommendations to improve estimation practices
     - Make software cost estimates more defensible when negotiating over project budgets
     - Improve software cost estimation ‘accuracy’

© 2012. All rights reserved.

Approach and Methodology -1

1. Two step data collection approach
   - On-line Survey
   - In-depth Interviews
2. Identify weaknesses and potential fixes
   - Some of these may be out of our control and more difficult to change
3. Identify strengths that we can build on
4. Identify areas for improvement
5. Recommend a set of cost metrics
   - Agency metrics to measure software cost and schedule over runs
   - Center level metrics to support software cost estimating
6. Develop Agency wide guidelines and work aids for software cost estimation (tailored at center level)

© 2012. All rights reserved.

Approach and Methodology - 2

The primary focus this year was to document the current software cost estimation practices and identify recommendations

1. **On-line Survey** (we are reporting these results today!)
   - Probes for basic activities primarily within the software community across all centers. For example, do you use cost models, etc.
   - Include all members of software community who develop, review, or support estimates at all centers
     - Dryden was not included because no names were submitted by the SWG lead
     - Include system level people who work with software intensive systems
2. **In-Depth Interviews**
   - Objective
     - Document what happens to cost estimates as they move up and out through the organization
     - Collect detailed descriptions of key software estimation practices

       Especially how they develop their BOE
   - Detailed interviews conducted at
     - GSFC, GRC, MSFC, JSC and IPAO

© 2012. All rights reserved.

Approach and Methodology – 3

Survey Instrument Example

Estimation Practices

14. Do you have a documented procedure for estimating software cost? (Required)
    - ○ Yes
    - ○ No

15. (Required)

	Always (90-100%)	Most of the time (75-90%)	Some of the time (40-75%)	Rarely (10-40%)	Never (0-10%)	N/A
If you have a documented procedure for estimating software cost, how often do you follow the procedure?	○	○	○	○	○	○

16. Estimation Practices (Required)

	Always (90-100%)	Most of the time (75-90%)	Some of the time (40-75%)	Rarely (10-40%)	Never (0-10%)
Do you use a functional decomposition (or similar technical	○	○	○	○	○
Do you estimate schedule?	○	○	○	○	○
As part of your cost estimate, do you deliver an integrated technical, cost, and schedule breakdown for the software task or product?	○	○	○	○	○
Is a structured breakdown of work used to estimate effort/cost and schedule? Usually called a WBS or work breakdown structure.	○	○	○	○	○
Do you use more than one method when coming up with a cost estimate?	○	○	○	○	○
Do you use historical data for high-level analogies?	○	○	○	○	○
Do you use industry wide software cost models (e.g. SEER-SEM, COCOMO)?	○	○	○	○	○
Do you use locally developed parametric software cost models?	○	○	○	○	○
Do you use a bottom-up estimate	○	○	○	○	○

© 2012. All rights reserved.

Survey Sample Size

Center	Number Responses
ARC	8
GRC	5
GSFC	23
JSC	3
KSC	9
MSFC	4
SSC	9
LaRC	5
JPL*	27
Total	**93**

- Target populations were members of the NASA software community who are directly involved with generating or reviewing software cost estimates
  - These were identified as the go-to people for software cost estimates at their center by the Software Working Group leads
- Responses were distributed across the centers and provided a 'good' representation of NASA software cost estimation practices
  - But this is not a formal stratification
- Survey achieved a 70% response with some centers hitting 100%, which was above the 66% target

JPL had 54 responses of which 27 were randomly selected from "roles with lower response rates"

© 2012. All rights reserved.

Respondent Characteristics

- Respondents provided a good mix between technical and management staff

Role	Percent of Respondents
Software Manager/Lead	55%
Line Manager	25%
Software Cost Estimator	6%
Software Systems Engineer	5%
Project/Systems Level	7%
Other	3%

- The actual distribution of the software related roles (or other characteristics) is, at best, only partially known across NASA
- Therefore a formal stratification is not very feasible
- However, based on data from JPL and the OCE Software Survey, the distribution of the respondents does not appear unreasonable

© 2012. All rights reserved.

Sample Project Characteristics

- The estimation practices described in the survey results are distributed across the key characteristics (Domain, SW Class, Team Size, Lifecycle Phase)

Software Domain	Percent of Respondents
Flight Software	30%
Instrument Flight Software	5%
Ground Software	65%

Software Class	Percent of Respondents
Class A	9%
Class B	34%
Class C	38%
Class D	8%
Don't Know	11%

Lifecycle Phase	Percent of Respondents
Early Concept Design	14%
Proposal Estiamtes	25%
System Requirements	13%
Preliminary Design	16%
Recurring Project Estimates	33%

Software Team Size	Percent of Respondents
Small (1- 2)	18%
Medium (3-5)	39%
Large (6-10)	11%
Very large (>10)	32%

- Some concern that 11% do not account for Software Class when making an estimate (most of these are probably proposal estimates)

© 2012. All rights reserved.

Software Cost Estimation Environment

- Requirements immaturity is a fact of life with most early estimates
- However, even at PDR only 43% report that requirements are well defined at least most of the time
- Software requirements mature later in the lifecycle than hardware

Are software requirements well defined when making the estimate?	
Response	**Percent of Respondents**
Most of the Time to Always (75-100%)	28%
Some of the time (40-75%)	36%
Rarely to Never (0-40%)	35%

- Requirements volatility is a key issue to address

  It's possible to manage this volatility as a risk!

  Procedures and standards could make this easier and more routine.

© 2012. All rights reserved.

Software Cost Estimation Environment (continued)

How often is the software budget or allocation set before you determine the scope of the job?	
Response	**Percent of Respondents**
Most of the Time to Always (75-100%)	55%
Some of the time (40-75%)	27%
Rarely to Never (0-40%)	18%

Have you felt what you perceive to be inappropriate pressure to alter your software cost estimates?	
Response	**Percent of Respondents**
Most of the Time to Always (75-100%)	34%
Some of the time (40-75%)	33%
Rarely to Never (0-40%)	34%

Is it your perception that your best (recommended) estimate is included in the system estimate or proposed budget?	
Response	**Percent of Respondents**
Most of the Time to Always (75-100%)	48%
Some of the time (40-75%)	33%
Rarely to Never (0-40%)	19%

- There is an apparent inconsistency in the responses
  - 48% say best estimate gets into the budget
  - While 55% report the budget is set before the scope can be determined and 34% report there is frequently inappropriate pressure to alter estimate in some way
- The In-depth interviews revealed that they frequently negotiate a budget that they can accept, but at a much higher risk

© 2012. All rights reserved.

How Does NASA Estimate Software?

All of the following cost estimation practices should be considered Best Practices as they show up repeatedly in cost handbooks (NASA, GAO, JPL)

1. *The WBS is the primary 'tool' that provides structure for one's entire plan, providing consistency and decreasing the likelihood of forgotten items*
   - 52% reported using a Work Breakdown Structure (WBS)
2. *Models and data make our estimates repeatable, more accurate and defensible and multiple methods improve estimate robustness*

Use of supporting Analogies and Model-based estimates

- 25% use more then one method
- 65% use top level analogies (but only 37% capture actuals!)
- 26% use cost models (much higher for large flight SW tasks)
  - Increase from 9% in 1990 (JPL Study)

Basic Grass-roots estimates

- 57% develop a bottom-up estimate of some kind

Results shown above are for responses indicating that the practice is performed at least "most of the time" (>75%)

© 2012. All rights reserved.

How Does NASA Estimate Software? (continued)

3 *Many estimators generate comprehensive estimates incorporating schedule and technical breakdowns as part of their cost estimate*

- 55% deliver an integrated technical, cost, and schedule breakdown for the software task or product as part of their cost estimate
- 85% estimate schedule and effort/cost

4. A significant population within the NASA software community size their systems and code prior to cost estimation (this is required for cost model input)

- 59% size their software systems

And of those who do size their systems…

- 41% estimate Source Lines of Code (SLOC)
- 26% estimate Modules
- 18% estimate Function Points

Results shown above are for responses indicating that the practice is performed at least "most of the time" (>75%)

© 2012. All rights reserved.

How Does NASA Estimate Software? (continued)

5. *Failure to include risk and potential mitigations in our cost estimates can cause under estimation and increases the likelihood of cost overruns*
   - 48% of respondents specifically identify risks
   - 52% adjust their estimates for significant risks
   - 23% incorporate probability or statistical cost information when developing a cost estimate
   - 18% report probabilistic information with their estimate (e.g. an S-curve, cost range, or percentile)

Results shown above are for responses indicating that the practice is performed at least "most of the time" (>75%)

© 2012. All rights reserved.

Where Do the Software Estimates Go?

- We need to improve our understanding of what happens to our software cost estimates as they move into the larger project environment
  - Results indicate we need to work more closely with the costing community along with systems engineering community
  - Results clearly show that as a software project moves from the proposal stage to becoming a project, that software is likely to:
    - be demoted to a lower level in the WBS/Org Structure
    - have the software 'best' estimate be over ridden by the project or other parts of the organization

Question	Proposal	Project at PDR
Is software a line item in the final proposal or project cost estimate?	54%	22%
How often is the software budget or allocation set before you determine the scope of the job?	36%	30%
Is it your perception that your best (recommended) estimate is included in the system estimate or proposed budget?	53%	33%

© 2012. All rights reserved.

Preliminary Recommendations

This is a tentative list of recommendations which will be finalized after the analysis and report are completed.

All NASA software organizations should:

1. Have a defined software cost process which makes their estimation practices repeatable and enables incremental improvement
   - 90% of respondents follow the process when one exists
2. Educate center software cost estimators to use best practices
   - Already in process through the NASA Software Management Class and JPL SW Cost Estimation Class which rotates around the centers
3. Maintain a local database of their estimates and actuals
   - Only 37% of those polled capture actuals
   - Don't rely solely on the accounting system to get actuals
4. Require a minimum of 2 estimates for Class A and B software… also for Class C software if possible (easier if an established processes exists)
   - Recommend model use and the development of probabilistic cost estimates reflecting risk
5. Monitor and evaluate cost and schedule growth across software tasks and projects

© 2012. All rights reserved.

Conclusion

- The NASA OCE Software Working Group is taking steps to change the NASA software engineering culture from the grass roots level of the organization so that our engineering managers fully accept that cost estimation is a part of their job.
  - Each of the Best Practices are being performed individually somewhere in at least one NASA Center and many are being performed at multiple Centers.
  - But nowhere are they all being performed consistently at any one center.
- Next steps are
  - Complete the analysis and finalize the recommendations
  - Next year provide support to the centers in implementing the recommendations
    - Especially to establish databases and develop a cost model capability
- Possible future steps – Have you considered doing this across all engineering disciplines

© 2012. All rights reserved.