How Does NASA Estimate Software Cost? Summary Findings and Recommendations

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

Presentation to
NASA Mission Software Steering Committee
November 13, 2012

© 2012. All rights reserved.
Motivation

- Software cost growth rates of 50% to over a 100% are common
  - JWST and MSL are recent examples that exceed these numbers
  - There are internal and external factors that drive these numbers
- Many of these causes can be addressed through Estimation Best Practices
  - Especially when combined with an overall software process improvement program
    - At JPL we have documented a reduction in cost growth of 2/3rds for tasks that are identified as having greater than 80% process compliance
- While teaching at other centers in FY11 it was obvious that there are practitioners and various improvement activities occurring throughout the agency so we can learn from one another and build on each other's successes
  - Increased use of parametric cost models (SEER-SEM and COCOMO)
  - Beginning to develop software cost metrics repositories
  - At one center the Costing Branch/Office actually has a highly co-operative relationship with software managers
- There will clearly be differences in the details of how each center moves forward but the basic elements are the same and there are many similarities in our work.
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’
Two step data collection approach

- **On-line Survey** (93 completed surveys)
  - Probes for basic activities primarily within the software community
    - Stratified across centers, roles and software
- **In-depth Interviews** (conducted 39 interviews)
  - Document what happens to cost estimates as they move up and out through the organization
  - Population
    - Project Managers, Proposal Managers, Systems and Subsystem Managers, Center Cost Analysts, Line Managers, Software PDLs, Software Cost Analysts (SCA)
    - Collect detailed descriptions of key software estimation practices
      - Especially how PDL’s and SCA’s develop their BOE
      - Estimation of the appropriate level of software assurance
  - Detailed interviews completed at GSFC, GRC, MSFC
  - Detailed interviews planned for JSC and IPAO
Software Cost Estimation Environment

- Requirements immaturity is a fact of life with most early estimates
  - Only 28% reported requirements were well defined most of the time
  - Even at PDR only 43% report that requirements are well defined most of the time

- The In-depth interviews revealed that Line Managers, PDL’s and software cost estimators frequently negotiate a budget that they can accept, but with higher risk tolerance. The on-line survey results indicated that
  - Less than half reported they get their best estimate into the budget while the other half report the budget is set before the scope can be determined
  - 34% report there is frequently inappropriate pressure to alter estimate in some way.

- 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.

  On-line survey results shown are for responses indicating that the practice is performed at least “most of the time” (>75%)
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, AFRL)

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 than one method
     - The primary methods used are bottom-up or high level analogies
       - 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)
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
     – The weakness appears to be lack of formally integrating the cost and schedule with the design.

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 most use Source Lines of Code (SLOC)

Results shown above are for responses indicating that the practice is performed at least “most of the time” (>75%)
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)
- A significant weakness in this area was also indicated in the interviews.

Results shown above are for responses indicating that the practice is performed at least "most of the time" (>75%)
7. We need to improve our understanding of what happens to our software cost estimates as they move into the larger project environment

• 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

• Results indicate we need to work more closely with the costing community along with systems engineering community
In-Depth Interview Results for Key Roles in the Software Community

1) Software Cost Analysts (SCA)
2) Product Development Leads (PDL)
3) Line Managers
Software Cost Analysts (1)

- Key results or findings
  - In general, they are the primary estimator for early lifecycle in-house software projects/tasks including proposals but they are not involved afterward once a PDL has been assigned
  - They are very rarely involved with estimating contracted software
  - They have a better defined repeatable process than the line managers and PDL’s, but there are still areas for improvement
  - 50% reported regularly using multiple estimation methods
  - 75% reported regularly using either the SEER-SEM or COCOMO cost model
  - Most respondents report having access to some form of historical data, However, it seems that any ‘databases’ are relatively new and/or sparse.
    - Logical LOC and effort are primary metrics
Software Cost Analysts (2)

- **Key results or findings (continued)**
  - All respondents reported that their current BOEs are effective
  - Reuse assumptions are never validated and rarely challenged
  - Surprisingly excessive external pressure to low-ball estimates was not identified as an issue
- Either the software budget is too small of a percent of the total to be a major driver for the project or the cost iterated until an agreement is reached. This is achieved through a combination of descoping and increased risk tolerance.
Product Development Leads (PDL)

- **Key results or findings**
  - PDL’s typically do not come onto the task until account numbers show up
  - The majority of PDL’s primarily have a review, approve, negotiation function when setting the budget
    - They perform estimates as inputs into the negotiation process
  - Estimation methods reported by PDL’s vary widely across centers
    - PDL’s always use bottom-up estimation methods
      - 50% reported using multiple methods some of the time
  - BOE contents vary widely across PDL’s because their BOE’s are driven by the project
  - Only 60% reported their BOE’s were effective most of the time
  - Virtually all PDL’s reported that their projects were on budget
Line Managers

• Key results or findings
  – Primarily review and concur s/w cost
  – Software cost estimates are assessed rather informally, mostly via analogies and taking into account the experience of the estimator
  – All LM report that they believe that the initial estimates coming out of their Branch/Division are good.
    • However, this does not mean they are able to successfully defend their estimates.
  – In order to improve most line managers identified the need for greater consistency (‘standardization’) including documenting a costing process, access to tools and data and improved BOE’s.
  – Most have one “go to” cost estimator and believe that this is barely sufficient, and most recognize that they have no backup for these folks
  – 2/3 do not have documented processes for software cost estimates
    • Consistency is maintained by having all estimates reviewed by the “go to” cost estimator
  – Most are starting to compare cost/schedule actuals against the original estimates, though not in a standard way and its not accessible to others
Cross Cutting Questions

• There were three areas of concern that were probed across multiple roles
  – How Basis of Estimates are currently developed and documented
  – Training and growing good software cost estimators
  – Estimation of the desired Software Assurance support early in the life-cycle

• Questions were asked in both the on-line survey and the in-depth interviews that addressed these areas
  – The results are not entirely consistent
Basis of Estimate (1)

- **Description of role and respondents**
  - Both PDL’s and software cost estimators were asked about the contents of their basis of estimate
  - The set of items identified were based on what are considered software cost estimation best practices

- **BOE’s should include**
  - WBS (work breakdown structure)
  - WBS dictionary
  - Effort Estimates with supporting assumptions and analogies
  - Planning Parameters or supporting lower level estimates (e.g. Lines of code)
  - Supporting Model Estimates and Analogies
  - Schedule
  - Procurements
  - Cost estimates
  - Significant Cost and Risk drivers
  - Risk List/Issues/Known Liens
Basis of Estimate (2)

• PDL’s and Software Cost Analysts
  – Both develop BOE’s and both have some content that addresses most areas
  – PDL’s primarily respond to the project specifications and report that their BOE’s are frequently a bottom up estimate in spreadsheet sometimes with minimal supporting details
  – The SCA’s have greater consistency and deliver more of a complete package
  – Both report weaknesses in
    • documenting planning parameters and risks
    • model use
  – Major differences between the two are
    • Only 60% of PDL’s report their BOE’s are effective while all SCA’s report their BOE’s are effective
    • Only 50% of PDL’s report a SW WBS while it is standard practice for the SCA’s
    • Only a few PDL’s systematically identify specific risks as part of their BOE while the majority of SCA’s do identify the associated risks
Basis of Estimate (3)

• **Recommendations**
  – All line organizations should have a required BOE template
  – Improve cross fertilization of methods between the SCA’s and the PDL’s as they appear to often have different capabilities
  • Currently the proposal estimates are done by the SCA’s and the PDL’s typically take over when a cost account appears.
  • One approach is that the SCA’s do the proposal estimates (as they currently do at most centers) and then provide a backup estimate for the PDL’s when they do their bottom up estimates at project start.
Developing Software Cost Estimators (1)

• **Description of role and respondents**
  – Both line managers and software cost estimators were asked
    • What attributes of your organization’s best software cost estimator makes that person the best?
    • How long does it take to train a good software cost estimator?
  – These responses were very consistent across all respondents

• **Desired Attributes or Characteristics**
  – Detail oriented but can keep eye in the big picture
  – Team player
  – Must be able to defend their estimate under pressure (not a push-over) while maintaining composure
  – Recognize personal weaknesses and seek necessary help from team
  – Willing to ask questions and sometimes look stupid
  – Good Communication skills
  – Some leadership skills (so can get people to do things they do not want to do)
Developing Software Cost Estimators (2)

• **Training a good software cost estimator**
  
  - Primarily they need to be grown and mentored
  - Need to plan ahead and grow people so they have significant software development experience.
    
    • A specific number of years was not specified (assume it is at least 10 years)
  - Requires breadth of experience - Experience with different types of software
    
    • At least the different types of software developed in the branch or division
  - Perform approximately 5 cost estimates mentoring with someone else who has cost experience.

• **Recommendation**
  
  - Line organizations need to start identifying and grooming their go to software cost estimators
  - The SCA’s need to be able to do various types of estimates and be able to organize and maintain the necessary tools, data and processes
Estimating Software Assurance Needs

Description of role and respondents

- All 39 respondents were asked the same set of software assurance related questions
  - Do you consider the cost for Software Assurance (Safety, Reliability, Quality)?
    - If so, in what way do you consider /estimate SW Assurance?
    - If not, are you aware if it is estimated somewhere else? If so, where-who is responsible for this estimate?

Key Findings

- As expected virtually no software personnel carry a Software Assurance budget allocation within their budgets.
- The on-line survey indicated that a third of the respondents rarely to never include support services and of the half of the respondents who do consider support services it was not clear that they included S&MA in their estimate

Recommendations

- Software Assurance organizations and Software Development organizations need to work together when developing cost estimates.
- Software Assurance organizations need to show value and to find a way to collect and use actuals from previous projects
Recommendations for Software Organizations

1. All software organizations should have a documented process
   - Process needs a standardized BOE Template with examples (important to be standardized as it makes estimates more defensible and enables archiving and reuse of BOE information)
   - At some point in the lifecycle software organizations should provide at least 2 estimates (Definitely by PDR) because of scope uncertainty

2. Encourage use of tools and data
   - NASA Costing Office already provides free access to SEER and Price software cost models
   - All software organizations should establish and maintain a historical DB

3. Each Software intensive branch or division should have a specific person who fulfills the software cost analyst role, which may be a part time role
   - This person also should also act as a bridge between the engineering and center costing organizations for best practices and access to tools and data (Infusion Agent)
Recommendations for NASA OCE SWG

1. Develop NASA wide examples and templates to support center development of
   - Cost process, tools, BOE templates, WBS templates
2. Educate center software cost estimators and PDL’s to use best practices
   - Already in process through the NASA Software Management Class and JPL SW Cost Estimation Class which rotates around the centers
   - Fund SEER-SEM Training
3. Bring together SW engineers from across the agency that have experience using SEER-SEM to determine recommended NASA-specific settings such as how to adjust the model to handle various classes of software and safety-criticality
4. Monitor and evaluate the effectiveness of the implementation of the study recommendations
   a) implementation and use of processes and templates
   b) cost and schedule growth across software tasks and projects

© 2012. All rights reserved.
Conclusions and Next Steps

- Next major steps are
  - Develop a draft tailorable NASA wide software cost estimation process with supporting tools, checklists, templates and guides
  - Identify infusion agents at the pilot centers
  - Pilot the process by providing support to the centers in implementing and infusing their centers version of the cost cost process
  - Provide regular status reports to the SWG and MSSC
  - Update the NPR and the Software Handbook as needed