PSM 2008 Workshop: Toward Integrating Systems Engineering and Software Engineering Estimation: Harmonizing COSYSMO and COCOMO

John Gaffney
301-721-5710 j.gaffney@lmco.com
Garry Roedler
610-354-3625 garry.j.roedler@lmco.com
Program Resource Center
Corporate Engineering and Technology
Lockheed Martin Corporation
July 16-17, 2008

Workshop Premise

It is highly desirable to integrate systems engineering estimation and software engineering estimation

Background/Observations

- This is the latest of a series of workshops on systems and software engineering estimation (COSYSMO and COCOMO) that have occurred over a number of years at PSM and in other venues such as the University of Southern California COSYSMO/COCOMO Workshops
- The COSYSMO systems engineering estimation model/tool and its risk and reuse extensions, the COSYSMOR tool developed by Lockheed Martin and endorsed in several venues including PSM 2007, and the COCOMO software engineering estimation model/tool have evolved somewhat independently, resulting in:
  - A disjunction of the two sets of cost drivers, e.g., COCOMO can represent the affect on cost of schedule compression, but COSYSMO can not
  - The need to relate and harmonize the tasks and the phases/portions of the product or system life cycle covered by each tool

Workshop Goals/Intended Products

1. Obtain some degree of consensus about what changes (additions/deletions/definitions), if any, should be made to the COSYSMO and COCOMO cost drivers
2. Obtain some degree of consensus about what the tasks or activities and life cycle phases in COSYSMO and COCOMO should be
3. Identify other concerns and problems relating to harmonizing COSYSMO and COCOMO and identify the next steps to be taken

Review/Level Setting: COSYSMO and COCOMO Model Form

Both COSYSMO and COCOMO are parametric models of the form $K=A*S^E*(\Pi D_i)$ *****,

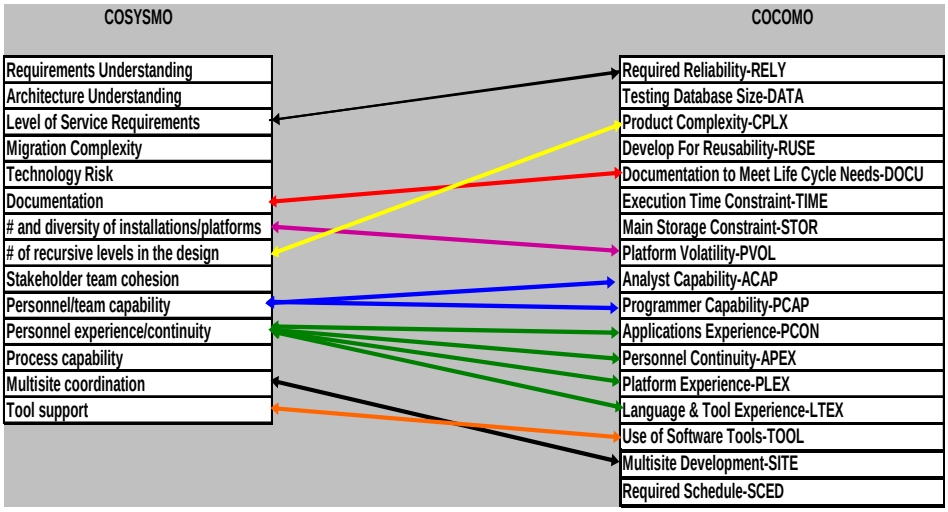
- Where: K is the cost/effort estimated for the project; S is size or scope and the cost drivers, D_i (i=1, 2, ...n) are selected for the project. The values for A, the productivity constant, and E are based on organizational experience
- The D_i are presumed to be mutually independent
- ΠD_i modifies the productivity (actually, unit effort) for the domain/organization, e.g., sonar software coded in C^{++}, for the particular situation,<product/project, process, personnel, tools>, being estimated

  * Note: *This form is basically the same as the activity-based model where typically only one or a small set of activities is included and cost driver values (e.g., D_i s) are not explicitly stated*

Review/Level Setting: Cost Drivers

- The cost drivers, the D_i, characterize attributes of: the product/project, the processes used to perform the task estimated, the personnel, the tools, and in the case of COCOMO at least one attribute of the processor, e.g., Main Storage Constraint
- Each driver value is selected with respect to the baseline's value, captured in the constant A (see previous page). "Nominal" for driver D_i is coded as D_i=1.0, i.e., the project being estimated does not differ (at least not appreciably) from the baseline with respect to the attribute characterized by D_i

COSYSMO and COCOMO Cost Driver Mapping



Arrows indicate corresponding COSYSMO and COCOMO cost drivers

Some Suggestions For Consideration

- COCOMO Drivers, Attribute Representations, that COSYSMO Needs
  - Does COSYSMO need a driver corresponding to Execution Time Constraint (TIME) ?
  - Does COSYSMO need a driver corresponding to Storage Constraint (STOR) ?
  - COSYSMO does need a a driver or some other way of representing schedule compression (SCHED)*
  - Does COCOMO need Migration Complexity ?
  - Does COCOMO need Technology Risk ?
- COSYSMO Drivers, Attribute Representations, that COCOMO Needs
  - Requirements Understanding
  - Architecture Understanding
  - Process Capability

*Note: The COCOMO SCHED driver values appear excessive, i.e., 1.14, 1.43; Another approach, e.g., estimate based on schedule vs. cost for a given size, may be better to represent the operative schedule/cost relationship

Backup

Parametric Cost Estimation

"Parametric techniques focus on the cost drivers, not the miscellaneous details. The drivers are the controllable system design or planning characteristics and have a predominant effect on system cost. Parametrics uses the few important parameters that have the most significant cost impact on the product(s), hardware or software, being estimated."

Source: Parametric Cost Estimating Handbook, sponsored by a joint Government/Industry Committee formed in 1994 to study ways to enhance the use of parametric cost estimating techniques

Activity-Based Cost Models

- Activity Based Cost (ABC) Models:
  - Estimate the costs for each activity or group of activities that compose project
    - Ideally, derived from the work break-down structure (WBS)
  - Enable the estimator to separately consider each activity in the process (e.g., software development process)
    - Provides an intellectual framework for considering the effect of changes (relative to past experience) such as: a new tool, a process change, different skill mix, etc.
    - Users need to identify all of the activities that compose the specific process whose cost is to be estimated
    - Relate to WBS and potentially more specific staffing
- Cost elements may be driven by:
  - Size of the product
  - Proportions (percents) of the cost (effort)
    - Examples: quality assurance, builds and controls, program office