Systems Engineering Cost Estimation by Consensus

Ricardo Valerdi  
Center for Software Engineering  
University of Southern California  
Los Angeles, CA 90089  
rvalerdi@sunset.usc.edu

Christopher Miller  
Software Productivity Consortium  
2214 Rock Hill Road  
Herndon, VA 20170  
Miller@software.org

Gary Thomas  
Raytheon - Intelligence & Information Systems  
P.O. Box 660023  
Garland, TX 75266-0023  
Gary_D_Thomas@raytheon.com
Abstract

A number of companies have developed models for estimating system resources for their projects. While these models are useful they are not available to industry or academia to validate or improve. Moreover, proprietary models are often narrowly focused on a single application domain.

A group of systems engineers and cost modelers from industry and academia have joined forces to develop a model, called COSYSMO, that will be available to the public in the near future. COSYSMO is the first openly available parametric cost model for systems engineering. The process used to develop the model has played a key role in the success of the model thus far. This collaborative effort has helped us ensure that the model (1) meets the needs of INCOSE members, (2) accurately estimates systems engineering resources, and (3) is applicable to multiple domains of systems engineering.

This paper is divided into four sections. First, it provides a rationale for the model. Second, it highlights recent challenges in developing an estimation model with an SE focus. Third, it describes the process used to develop the model and focuses on the aspect of consensus building. Finally, it outlines the main features of the prototype implementation of the model myCOSYSMO. A discussion of ongoing work is also provided together with a list of challenges that lie ahead.

Key words: model building, cost estimation, systems engineering estimation, COSYSMO, measurement & metrics
Rationale for the Model

Under the software Capability Maturity Model (CMM) paradigm, systems engineers established system requirements and allocated some to software. The software engineers built the software to the requirements then systems engineering integrated the developed software into the overall system solution and verified that the requirements were satisfied. The resulting project social structure looked something like Figure 1.

![Figure 1. Resulting Project Structure – CMM View](source: Boehm 2002. Used with permission.)

Under the integrated software engineering and systems engineering paradigm, or Capability Maturity Model Integration (CMMI), software has a seat at the center table. A project’s requirements, architecture, and process are collaboratively developed by integrated teams based on shared vision and negotiated stakeholder concurrence.

Over the past twenty years, software has developed a large body of knowledge related to software cost estimation that often has included significant enterprise investment in the collection and analysis of measurement data to support the software estimation process. While there are some good proprietary cost models available, the mostly widely used “open” model available in the public domain is COCOMO II (Constructive Cost Model).

A close examination of CMMI process areas (particularly for the staged representation) strongly suggested the need for the systems engineering function to estimate systems engineering effort and cost based upon a consistently provided organizational approach (important for CMMI Level 3) based upon past project performance measures related to size, effort and complexity. While it might be possible to achieve CMMI levels without a parametric model, an organization should consider the effectiveness and cost of achieving them using other methods that might not provide the same level of stakeholder confidence and predictability.

It became apparent that the systems engineering community had an opportunity to heavily leverage the excellent work done for software cost estimation by the University of Southern California Center for Software Engineering (USC/CSE). USC/CSE had the research methodology, knowledge, and tools based upon their recent COCOMO II development project to develop a Systems Engineering “emerging extension” to the current COCOMO II software cost model baseline. This “open” model and the consensus process used to develop it has been documented in *Software Cost Estimation with COCOMO II* (Boehm, et al. 2000). As a result, the Constructive Systems Engineering Cost Model (COSYSMO) was developed to meet this need and address the challenge of estimating Systems Engineering Cost. INCOSE and the USC/CSE Industrial Affiliates provided their systems engineering subject matter expertise in the development of this COCOMO II-derivative over the past three years by providing...
oversight, measurement data, peer review, and Wideband Delphi team membership. See the Table 1 for specific CMMI Process areas and related specific practices that can be supported by a model such as COSYSMO.

Table 1. COSYSMO Support for CMMI Estimation Relative Tasks

<table>
<thead>
<tr>
<th>Level 2</th>
<th>Project Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP 1.1</td>
<td>Establish a top-level Work Breakdown Structure to estimate the scope of the project</td>
</tr>
<tr>
<td>SP 1.2</td>
<td>Establish and Maintain Estimates of the Attributes of the Work Products and Tasks</td>
</tr>
<tr>
<td>• Size and Complexity Drivers; Estimation Models</td>
<td></td>
</tr>
<tr>
<td>• SP 1.4 Estimate the Project Effort and Cost for the Work Products and Tasks Based on Estimation Rationale, Estimation Models; Use Historical Data</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project Monitoring and Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP 1.1</td>
</tr>
<tr>
<td>• Monitor Project's Cost and Expended Effort</td>
</tr>
<tr>
<td>• Monitor Attributes of the Work Products and Tasks</td>
</tr>
<tr>
<td>• Monitor Resources Provided and Used</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 3</th>
<th>Integrated Project Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP 1.2</td>
<td>Use the Organizational Process Assets and Measurement Repository for Estimating and Planning the Project’s Activities</td>
</tr>
<tr>
<td>SP 1.4</td>
<td>Manage the Project Using the Integrated Plans</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 4</th>
<th>Organizational Process Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP 1.5</td>
<td>Establish and Maintain the Process Performance Model for the Organization’s Set of Standard Processes, Cost Estimation Model</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quantitative Project Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP 2.1</td>
</tr>
<tr>
<td>• Cost Estimation Attributes; Data Collection; Continuous Model Calibration</td>
</tr>
<tr>
<td>SP 2.2</td>
</tr>
<tr>
<td>• Cost Estimation Attributes; Data Collection; Continuous Model Calibration</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 5</th>
<th>Organizational Innovation and Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>A parametric approach to Systems Engineering Cost Estimation is an innovative leap from where the state of the practice is today, but alone does not qualify for Level 5 OID compliance.</td>
<td></td>
</tr>
</tbody>
</table>

Challenges in Measuring Systems Engineering

There are many challenges facing the development of a successful systems engineering cost parametric model. As with any measurement, the entity being measured must be identified. In this case, the entity is systems engineering. Defining systems engineering for the purposes of measuring it, can be quite elusive. Sheard (1996) revealed that there are different kinds of systems engineering across industry. We have even witnessed varying definitions of systems engineering within organizations. For matrix organizations supplying personnel to systems engineering contracts it is common for the contract to define the scope of the systems engineering activities, a scope which is rarely consistent across contracts and customers. The existence of multiple definitions of systems engineering presents a major obstacle in establishing meaningful measures, and subsequently meaningful parametric models. Measuring the macro activity (i.e., systems engineering) is essentially meaningless unless the different roles and activities of the systems engineers are performed consistently across projects and across divisions.
The fundamental assumption required when developing a prediction model is that past performance is the best indicator of future performance. Meaning that if we measure activity x for multiple iterations, then a base of performance data can be collected and analyzed to assist the prediction of the next iteration of activity x. This is assuming activity x is performed multiple times in a consistent way to make the data collected meaningful. The activities that comprise systems engineering vary per iteration. The underlying question that remains to be answered is whether or not this variation is manageable or whether it is so great, thus preventing the detection of signal through the noise is not possible. COSYSMO will only be able to answer this question when data has been collected and analyzed.

An additional challenge with quantifying systems engineering resides in the lack of tangible interim work products. Quite often very valuable systems engineering work involves the ability to identify system level issues. For example, a systems engineer may recognize that the evolving system architecture specification has potentially incompatible interfaces. Given that the specification is still a work-in-progress, the systems engineer simply identifies the issues and recommends to the program manager to initiate a trade study aligned with identifying and evaluating interface solutions for this component of the architecture. Without the systems engineer’s input this interface issue may not have been raised until the components failed to integrate, at which point the cost and schedule impacts of reworking the interface would be astronomical. Previous work has been done by others (Ernstoff and Vincenzini 1999) to categorize SE products but this categorization is specific to one particular hardware application domain and cannot be generalized for other SE domains.

When we try to quantify systems engineering in terms of capturing productivity (i.e., size/effort) data to incorporate into a parametric model, what size measure captures the amount of intellectual work performed by the systems engineer? Are number of issues raised a good indicator? Size of the interface? Complexity of the interface? Size of subsequent trade study? Size of the architecture specification? These types of scenarios were heard often during many of the COSYSMO working sessions. The working group recognized the need to define systems engineering for the purposes of developing this model.

After much debate, the COSYSMO working group converged on a definition and adopted two Systems Engineering standards: EIA/ANSI 632 and ISO/IEC 15288. The EIA/ANSI 632 standard provided a set of Systems Engineering activities that we could include in the model. The ISO/IEC 15288 standard provided a set of Systems Engineering life cycle phases that we could use as the primary structure for building the model. In order to use COSYSMO accurately, estimators will need to map their systems engineering activities to the activities in EIA/ANSI 632 and their system life cycle phases to ISO/IEC 15288. Identifying the similarities and differences will provide each estimator with an initial indication as to the usefulness of the model for their individual situation.

Model Development Methodology
The USC/CSE has many years of experience in developing parametric cost models. In addition to the COCOMO II model there have been other parametric models developed for specific needs including COCOTS (for commercial off-the-shelf software), CORADMO (for rapid application development), COQUALMO (for software quality), COPROMO (for productivity improvement), and COPSEMO (for phased schedule and effort). COSYSMO is the first model to focus on issues outside of the software domain and, as a result, is breaking new ground and facing new challenges.

The methodology used to develop the COCOMO Suite is comprised of seven steps that are outlined in Boehm (2000). These steps are (1) analyze existing literature, (2) perform behavioral analysis, (3) identify relative significance, (4) perform expert judgment via Delphi, (5) gather project data, (6) determine bayesian a-posteriori, and (7) gather more data and refine model.

The initial step implied in this methodology is that a stakeholder community has been identified and is willing to support the development of the model with financial resources and data to calibrate it. Once this step is complete the model can begin to take form.
Reaching Consensus

Steps three and four were the most influenced by members of the COSYSMO Working Group – a collection of industry experts that served advisory roles in the definition and scope of the model. As with most USC/CSE efforts, the requirements for a systems engineering cost model was levied by the CSE Corporate Affiliates. The COSYSMO development team and working group consists of approximately 15 people. The COSYSMO development community includes over 100 people involved in its meetings since its inception and approximately 50% of those individuals are routine participants that attend at least one working group meeting per year.

These working group meetings have served as focus groups to help reach agreement on definitions, scope of the model, identifying the applicable activities, and reaching consensus on the cost and size drivers and range of variation. The action items from these meetings helped the group develop an initial framework for the model and enabled the team to devise a migration path. Members of the working group have continued to collaborate on solidifying the definitions of the parameters while focusing on the most difficult drivers, the size parameters. Work is being done to address the challenges in sizing systems engineering work (Valerdi et al 2003). The fourth step in the model development consisted of gathering expert opinion from the Systems Engineering community using the Wideband Delphi technique. This technique has been identified as being a powerful tool for achieving group consensus on decisions involving unquantifiable criteria and has been successfully used in the development of other COCOMO Suite models. The Delphi process helped the working group reach consensus and validate the initial findings. So far there have been two Delphi rounds; the first round yielded 28 responses and the second round yielded 41. More than half of the respondents in round 2 also participated in round 1.

Successful Partnerships

The COSYSMO working group established a partnership with the INCOSE community via two technical board sponsored liaisons, collaboration with the Systems Engineering Center of Excellence (SECOE), and an alignment of working group activities. The liaisons are active members of the COSYSMO working group and report status back to the INCOSE community. The INCOSE Measurement Working Group hosts the COSYSMO working group each February to give the COSYSMO working group an opportunity to interface with the systems engineering community at large. These exchanges are critical to refine the model requirements and elicit feedback on the interim work products. This interaction has allowed the COSYSMO development team to understand how the definition of systems engineering activities are interpreted and used by their end users. The refinement process associated with each of the size and effort cost drivers has required extensive effort and rework during each session. Although each session has required a lot of effort, there have been great increases in the quality of the model’s underlying assumptions and definitions. On the surface many of the cost drivers will appear to be same as COCOMO II, however a close inspection of the definitions will reveal the significant refocusing that was made to ensure the model drivers addressed systems engineering effort.

This type of collaboration has had two main effects. The primary effect is to develop and deliver a valuable systems engineering cost model to the systems engineering community. The other effect is a sense of ownership of COSYSMO within the INCOSE measurement community. The presence at INCOSE has heightened the awareness of the effort with the systems engineering community and has spurred the interest of systems engineering experts. This domain exposure is a part of the model development methodology to increase the chances of model adoption and successful implementation.
Current pilot implementation of myCOSYSMO

The extensive participation from INCOSE and the numerous working group meetings has stirred the interest for COSYSMO. The myCOSYSMO tool was developed primarily to capture the most currently available COSYSMO baseline and to provide a relatively simple means of visualization of the model to all stakeholders. It is a MS Excel™ based workbook with a few simple Visual Basic macros embedded that primarily provide a means of navigation. The latest version of the tool (1.16) is available for downloading from: http://www.valerdi.com/cosysmo. Opening the myCOSYSMO file, the user can choose either a “Costing Mode” or a “Data Collection Mode” as shown in Figure 2. For the moment the focus is on SE Cost estimation. By clicking on the “SE Costing Mode” we arrive the “Table of Contents” worksheet. The Table of Contents worksheet is “Home Base” for the myCOSYSMO workbook and provides a means to quickly navigate about the 30-plus worksheets. There are some basic conventions that are consistently applied throughout myCOSYSMO:

- Click on grey buttons to activate the relevant destination worksheet(s)
- Return back to TOC from the grey button labeled “TOC” in upper left hand corner of each destination worksheet
- Grey fields mean user can input or potentially change the default values provided
- Formula worksheets are protected, but no password is required
- Extensive embedded notes are provided on all worksheets mirroring current COSYSMO descriptions, driver selection criteria, etc.

A straightforward SE Estimate process can be applied using myCOSYSMO as follows (iterate Steps 2-6 below as appropriate):
1. Understand the Problem and Risks, determine whether and how COSYSMO should be used
2. Document Assumptions and Requirement Sources
3. Initialize/Update Project Parameters including labor profiles/rates
4. Estimate SE Size using the four size drivers
5. Rate the fourteen Cost Drivers
6. Review Generated Effort Hours and Costs. See sample outputs in Figure 4.
7. Optionally, submit the estimate to your pricing function for precise dollar cost. See comments regarding the need for an additional export file interface worksheet that will be specific to your organization.
The ability to create a simple organizational database of past project measurement data will be key to the success of COSYSMO because it will enable reasonable estimates of systems engineering effort. This data is critical in understanding the capabilities of your local organization. Each organization can create their own database using the worksheet provided in both the SE Costing and SE Data Collection Modes referenced earlier.

The organizational database can also aid in creating a local calibration of the model. There is a worksheet that can be easily tied to this SE Data Repository worksheet that performs a linear regression in a manner very similar to that suggested by the COCOMO II book. The calibration accuracy criteria (PRED, RRMS, and MMRE) are also borrowed from standards commonly used by the software estimation community. The regression formulas are relatively straightforward (embedded notes are provided), but the real issue is availability of past local SE measurement data to be able to “calibrate” your local performance capability. Figure 5 is the screenshot of the worksheet that allows for local calibrations.

![Figure 4. myCOSYSMO Executive Cost Summary](image1)

![Figure 5. myCOSYSMO Local Calibration Worksheet](image2)
Customizing myCOSYSMO

In some situations, the COSYSMO model might lack a multiplicative cost factor that is important to a particular enterprise of interest. For example, an organization may have projects with significantly different security constraints which influence systems engineering effort and cost. In this situation, it might be necessary to add a new cost driver, SCON (Security Constraints) to the model. The organization would then have to develop a suitable rating scale for SCON and locally calibrate the parameter. Ideally they should use data from completed projects or in absence of such data rely on expert opinion using the Wideband Delphi process. The myCOSYSMO tool provides templates for the addition of up to three such effort multipliers at the bottom of the Team Factors worksheet shown in Figure 6.

In addition, there are template worksheets for adding up to three systems engineering size drivers. For example, an organization might decide to develop their Systems Engineering sizing either in terms of Function Points or Top Level Design Classes in conjunction with the four size drivers (worksheets 12a-12d) already base-lined within COSYSMO. In all cases, the organization must first determine whether a newly proposed cost driver affects systems engineering effort in a multiplicative fashion (update the Team Factors worksheet) or in an additive fashion (update one or more of worksheets 12e-f.). A new driver called Function Points or Top Level Design Classes can be added and its effects can be accounted for in the COSYSMO Cost Estimating Relationship.

Data collection & next steps

The success of COSYSMO largely depends on the quality and quantity of data received to calibrate it. This data will determine the depth and breadth of the model and will drive its impact to the systems engineering and cost estimation communities. The immediate next step is to continue the process of collecting cost data for completed projects. The rest of the parameters for the systems (Requirements, Interfaces, etc.) will be counted retrospectively. A data collection form is being developed to facilitate this process and ensure consistent interpretation of the size and cost drivers.

Once the data is collected and validated it will be used in determining the relative significance of the parameters and general form of the model. The historical projects will define a data-driven model which might disagree with previous assumptions. At this point the parametric model is combined with the driver weights obtained from the Delphi survey using Bayesian approximation techniques. The final product will be a working model with solid Cost Estimating Relationships (CERs).

It is our goal to provide a model that not only meets the needs of INCOSE members but is also accurate and applicable to multiple domains. This can only happen with the continued participation of the systems engineering community and INCOSE corporate members.
References


Acknowledgements
The authors would like to thank the members of the COSYSMO Working Group especially: Elliot Axelband (USC), Barry Boehm (USC), Jim Cain (BAE Systems), Michael Ernstoff, Gary Hafen (Lockheed Martin), Cheryl Jones (US Army), Paul Mohlman (the Aerospace Corporation), Don Reifer (REI/USC), Garry Roedler (Lockheed Martin), Evin Stump (Galorath), Marilee Wheaton (the Aerospace Corporation), and Steven Wong (Northrop Grumman). The working group has also received much support from INCOSE member companies and Corporate Advisory Board members including: Dave Hickman (The Aerospace Corporation), John Rieff (Raytheon), Paul Robitaille (Lockheed Martin), and Jim VanGaasbeek (Northrop Grumman).