Definition and Initial Validation of a Systems Engineering Reuse Model

Jared Fortune*
The Aerospace Corporation, El Segundo, CA

Ricardo Valerdi†
Massachusetts Institute of Technology, Cambridge, MA 02139

As system complexity increases or resource availability becomes constrained, a common occurrence with space systems, systems engineers are frequently asked to (or seek to) leverage previously developed systems engineering products to reduce cost, schedule, and risk. This research presents the definition and initial validation of a systems engineering cost model (COSYSMO 2.0) for estimating the effect of reuse on systems engineering effort and cost.

Nomenclature

R_N = number of New requirements
w_N = weight of New sub-category
R_A = number of Adopted requirements
w_A = weight of Adopted sub-category
R_O = number of Modified requirements
w_O = weight of Modified sub-category
R_D = number of Deleted requirements
w_D = weight of Deleted sub-category
R_T = number of Total requirements
w_Y = weight of {New, Adopted, Modified, Deleted} category
R_{TE} = number of Total Equivalent New requirements
R_M = number of Managed requirements
R_F = number of Designed for Reuse requirements
PM_{NS} = effort in Person Months (Nominal Schedule)
A = calibration constant derived from historical project data
k = {Requirements, Interfaces, Algorithms, Scenarios}
r = {New, Design for Reuse, Modified, Deleted, Adopted, Managed}
w_X = weight for "easy", "nominal", or "difficult" size driver
Φ_x = quantity of "k" size driver
E = represents (dis)economies of scale
EM = effort multiplier for the jth cost driver

I. Introduction

For over thirty years the U.S. Government Accountability Office (GAO) has reported that "realistic cost estimating was imperative to making wise decisions for acquiring new systems" and during this time, the GAO

* Member of the Technical Staff, 2350 E. El Segundo Blvd., El Segundo, CA 90245. Member.
† Research Associate, 77 Massachusetts Avenue, Cambridge, MA 02139. Member.

has cited cost estimation problems with nearly every major NASA and DoD-space system acquisition.[1] As the size and complexity of space systems continues to increase, the need for better cost estimation tools in every aspect of system development is essential. Among all system development activities, one particularly critical group with relatively immature cost estimation techniques is systems engineering. Historically, space systems have estimated the cost of systems engineering activities as a percentage of total program, hardware, or software costs; however, such approaches fall short by failing to consider the unique characteristics and challenges of the individual program in the cost estimate.[2] In 2005, the Constructive Systems Engineering Cost Model (COSYSMO) was developed in coordination with industry and academia to produce a more quantitative and justifiable systems engineering cost estimate by accounting for the drivers of the systems engineering activities.[3] Although COSYSMO is a valuable tool for parametrically estimating the cost of systems engineering and has been widely accepted by several aerospace contractors, it is not without limitations. One limitation that will be addressed in this paper is the ability to account for the impact of reuse on systems engineering.[4] Systems engineering reuse is the utilization of previously developed systems engineering products from previous efforts such as requirements, test plans, and interfaces. As system complexity increases or resource availability becomes constrained, a common occurrence with space systems, systems engineers are frequently asked to (or seek to) leverage previously developed systems engineering products to reduce cost, schedule, and risk. For example, the amount of systems engineering effort is frequently less for a backup ground station installation or the last satellite in a block of satellites than for the first. This research presents the definition and initial validation of a cost model for estimating the effect of systems engineering reuse by creating an extension to the existing COSYSMO model.

II. Model Evolution

In 2006, approximately one year after the publication of COSYSMO, practitioners noted that large errors were being observed between some model estimates and actuals.[5] Upon further review, these variations were discovered to be mostly attributable to programs that reused a significant number of previously developed systems engineering products. At the time, the model could not adequately handle reuse since an underlying assumption of the tool was that all aspects of system development and related systems engineering activities were "built from scratch". In other words, COSYSMO assumed that all systems engineering activities and resulting products would need to be completed as new, and no systems engineering products were reused. While this assumption did not affect projects that did not leverage any previously developed systems engineering products, it did affect estimates of projects that did reuse, frequently over-estimating (potentially significantly) the amount of effort necessary to complete the systems engineering activities. At the COSYSMO Working Group meetings the following year, accounting for systems engineering reuse was identified as the most critical area of COSYSMO research by representatives from over eight industry and academic organizations.[6] Given the substantial interest in developing a reuse extension for COSYSMO, several strategies were been proposed and the model methodology evolved over time as the scope and definitions of systems engineering reuse gained wider acceptance.

The initial methodology for a reuse extension was to revise the estimate of the size of the systems engineering project by applying four types of reuse to the Number of Requirements size driver.[7] This methodology divided the Number of Requirements size driver into four sub-categories: New, Adopted, Modified, and Deleted requirements (Table 1). A model user would provide system specific quantities for each of the applicable sub-categories. For example (elaborated on in section IV), a model user may evaluate a system to have 185 New requirements (R_N), 60 Adopted requirements (R_A), 25 Modified requirements (R_O), and 0 Deleted requirements (R_D). The number of total requirements (R_T) would be calculated by multiplying the quantity for each sub-category by a pre-determined weighting factor (w_Y; where y = {New, Adopted, Modified, Deleted}) and then summing the products across the four sub-categories, as shown in Eq. 1.

	Sub-Category	*Definition*
1)	New	Items that are completely new
2)	Adopted	Items that are incorporated unmodified
3)	Modified	Items that are reused but are tailored
4)	Deleted	Items that are removed from a system

Table 1. Definitions of four sub-categories for the Number of Requirements size driver to account for reuse.

$$R_T = \left(w_Y \times R_Y \right) \tag{1}$$

This methodology would be repeated across each of the Easy, Nominal, and Difficult rating categories for the Number of Requirements size driver. The outcome of these calculations would be a revised value for the Number of Requirements size driver, termed Total Equivalent New Requirements (R_{TE}), which would replace Number of Requirements in the model, as shown in Eq. 2.

$$R_{TE} = (w_N \times R_N) + (w_A \times R_A) + (w_O \times R_O) + (w_D \times R_D) \tag{2}$$

The aim of this methodology was to reduce the number of requirements counted by COSYSMO by the amount of requirements to be reused. Although this methodology created a reasonable approach for accounting for reuse in the model (later strategies would build upon it), it was the first attempt at such a capability; however, because a consensus could not be reached on the appropriate number and scope of reuse categories, this approach did not receive full buy-in with the industrial community as the acknowledged approach for reuse in COSYSMO. The significant contribution from this methodology was the categorization of non-new requirements as modified, reused, or deleted.

After the publication of the initial reuse methodology and categories described above, a systems engineering organization applied it in an industrial setting.[8] The outcome of this application indicated that while the methodology was able to account for the effect of systems engineering reuse and did improve the estimation power of the model, four reuse categories were inadequate at capturing all types of reuse. Specifically, an additional category was needed to capture instances where systems engineering products are reused without modification or testing (e.g. when a system integrator incorporates a product from a subcontractor). As a result, a revised methodology was developed that included a fifth reuse category (defined in Table 2) to address instances of "managed" reuse (R_M). For the organization that applied this methodology, the five reuse categories improved the estimation power of COSYSMO.

While validating this revised approach among systems engineering organizations, several issues were raised. First, the five categories fail to account for instances of "design for reuse", when an upfront investment has been made to configure a systems engineering product so that it is reusable in anticipation of greater benefits throughout the life cycle. Second, the categories are only being applied to a single COSYSMO size driver (Number of Requirements). Although the Number of Requirements driver contributes a significant amount to the estimation power of the model, reuse has an important effect on the other drivers as well.[3] Lastly, more reuse categories provide additional explanatory detail to the model, but also increase the difficulty of implementation. To address these issues, a third methodology was developed, presented in Table 3. This methodology expanded the number of reuse categories to six and established a structure of prime and sub-categories.

	Sub-Category	*Definition*
1)	New	Items that are completely new
2)	Modified	Items that are inherited, but are tailored
3)	Adopted	Items that are incorporated unmodified, also known as "black box" reuse
4)	Deleted	Items that are removed from a system
5)	Managed	Items that are incorporated unmodified and untested

Table 2. Definitions of five sub-categories for the Number of Requirements size driver to account for reuse.

This third methodology addresses each of the issues previously raised. Although "design for reuse" was known to occur, previous methodologies did not explicitly call it out as a separate reuse category, it was grouped into the New category. With a separate

	Prime-Category	*Sub-Category*	*Definition*
I)	New		Items that are completely new
i)		Design for Reuse	Items that require an additional upfront investment to improve the potential reusability
II)	Modified		Items that are inherited, but are tailored
ii)		Deleted	Items that are removed from a system
III)	Adopted		Items that are incorporated unmodified (also known as "black box" reuse)
iii)		Managed	Items that are incorporated unmodified and with minimal testing

Table 3. Definitions of six sub-categories for the four size drivers to account for reuse.

Design for Reuse (R_F) category, the upfront investment for developing a reusable systems engineering product can be accounted for, which will typically exceed the cost of developing a new, non-reusable product, but may produce greater benefits (cost or effort savings) throughout the life cycle when reused. The prime/sub reuse category approach enables varying degrees of utilization of the reuse model. Some organizations expressed interest in applying only a few reuse categories, while others expressed interest in more categories and greater levels of detail. This approach is intended to capture three major, or prime, categories of reuse (New, Modified, and Adopted) as well as the three minor, or sub, categories of reuse (Design for Reuse, Deleted, and Managed). The number of Total Equivalent New Requirements for a system with reuse can be calculated with Eq. 3, a modified version of Eq. 2, which includes the six sub-categories.

$$R_{TE} = (w_N \times R_N) + (w_F \times R_F) + (w_O \times R_O) + (w_D \times R_D) + (w_A \times R_A) + (w_M \times R_M) \quad (3)$$

The methodology described above applies not only to the Number of Requirements size driver, but also the other three COSYSMO size drivers. Similar versions of the Total Equivalent New Requirements equation (Eq. 2) are applicable to the Number of Interfaces, Number of Algorithms, and Number of Operational Scenarios. The derivation and application of these equations are further discussed in section III.

An organization can use any or all of the six reuse categories when performing an estimate, depending on the particular circumstance. While more categories will likely produce a more comprehensive estimate, fewer categories will still enable accounting for some of the effect of reuse. These six reuse categories, discussed further in the following section, are the foundation for COSYSMO 2.0.

III. Model Definition

COSYSMO is part of the Constructive Cost Model (COCOMO) family of parametric cost models. Similar to COSYSMO and its reuse extension, models such as COCOTS and COPLIMO have been developed as extensions to COCOMO to better account for different types of software reuse[9]. Models such as COCOTS, COPLIMO, and COSYSMO 2.0 extend the capabilities of their predecessors, while maintaining similar mathematical form. As a result, the number and scope of the size and cost drivers of COSYSMO remain the same in COSYSMO 2.0—the six reuse categories are added as an extension to the size drivers, but have no effect on the estimate if no reuse values are inputted. The operational concept of COSYSMO 2.0 is illustrated in Fig. 1. Aside from the application of the reuse categories to the size drivers, the concept is the same for both models.

In COSYSMO, the user identifies the number of easy, nominal, and difficult requirements, interfaces, operational scenarios, and system-specific algorithms (the four size drivers). Using these as inputs and taking historical data into account, the model calculates the expected number of person-months needed to complete the systems engineering activities for the particular system of interest.[3] For situations where reuse is significant, some amount of the systems engineering activities will not need to be completed and the expected amount of time to complete the remaining systems engineering activities should be less than what was estimated in COSYSMO. In COSYSMO 2.0, the estimate of system size is modified according to the weights of the reuse categories. For any requirement, interface, operational scenario, or system-specific algorithm being reused, the user rates it as easy, nominal, or difficult, and determines which category of reuse is appropriate. The weight associated with each category of reuse is then multiplied across each of the three rating levels for each of the four size drivers. The result of this calculation is an adjusted estimate for project size, typically less than previously estimated by COSYSMO, and since reuse is accounted for, more representative of the expected project effort.

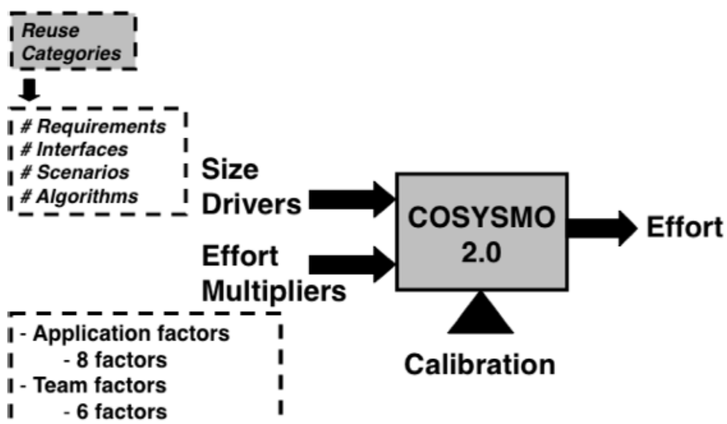

Figure 1. COSYSMO 2.0 Operational Concept

The COSYSMO 2.0 operational equation is presented in Eq. 4. The difference between the COSYSMO and COSYSMO 2.0 operational equation is the addition of the second summation and the second weight parameter, which collectively account for the effect of reuse. The second summation aggregates the weight of each reuse category multiplied across each size driver. The second weight parameter is the weight of each of the reuse categories. The "New" category has a corresponding weight of 1.00, since there is no reuse associated with it. The Modified, Deleted, Adopted, and Managed categories have weights less than 1.00 since there is some amount of reuse associated with them; a product with these levels of reuse should require less effort than a new, non-reused product. The Design for Reuse category has a weight greater than 1.00 since some first-time investment needs to be made to make something reusable; however, subsequent uses of a systems engineering product that was designed for reuse should be rated according to the expected reuse benefit and a weight less than 1.00.

$$PM_{NS} = A \cdot \left[\sum_k \left(\sum_r w_r \left(w_{e,k}\Phi_{e,k} + w_{n,k}\Phi_{n,k} + w_{d,k}\Phi_{d,k} \right) \right) \right]^E \cdot \prod_{j=1}^{14} EM_j \tag{4}$$

PM_{NS} = effort in Person Months (Nominal Schedule)
A = calibration constant derived from historical project data
k = {Requirements, Interfaces, Algorithms, Scenarios}
r = {New, Design for Reuse, Modified, Deleted, Adopted, Managed}
w_x = weight for "easy", "nominal", or "difficult" size driver
Φ_x = quantity of "k" size driver
E = represents (dis)economies of scale
EM = effort multiplier for the j^{th} cost driver

(4)

COSYSMO 2.0 accounts for reuse in the size drivers rather than in the cost drivers for three reasons. The first is that the four size drivers represent the majority of the explanatory power of the model. In other words, the variability in the COSYSMO data is mostly explained by the size drivers. By accounting for reuse in the size drivers instead of the cost drivers, the effect of reuse will sufficiently influence the resulting estimate. Another reason for reuse accounted for in the size drivers is that, to a limited degree, "reusability" is already accounted for in COSYSMO as part of the Level of Service Requirements cost driver. To avoid any confusion in double counting for the effect of reuse with multiple and/or overlapping cost drivers, the reuse extension is limited to the size drivers. By keeping the number and scope of the size and cost drivers the same in both COSYSMO and COSYSMO 2.0, and having reuse be an extension to the model, backward compatibility between both models and their data is maintained. A third reason for implementing the reuse weights on the size drivers is that they are continuous variables, as opposed to the cost drivers which are discrete variables. This provides the necessary level of fidelity and sensitivity to capture the impact of reuse on systems engineering size since there is a wider range of possible values that can exist to represent reuse. One of the hypotheses of COSYSMO is that:

> *A combination of the four elements of functional size in [the model] contributes significantly to the accurate estimation of systems engineering.*[3]

In other words, for a nominal project, the four size drivers can adequately represent the amount of equivalent person-months required for a systems engineering project. COSYSMO 2.0 follows a similar hypothesis:

> *Each COSYSMO size driver can be further decomposed into New, Design for Reuse, Modified, Deleted, Adopted, and Managed categories of reuse each with corresponding rating scales and weights, and function as accurate predictors of equivalent size.*

Following that the COSYSMO hypothesis is supported, the COSYSMO 2.0 hypothesis proposes that the application of the six reuse categories across all four size drivers adequately captures the effect of reuse on the amount of expected systems engineering effort.

IV. Initial Model Validation

After the COSYSMO 2.0 cost estimating relationship (shown in Eq. 3) had been defined, this research turned to conducting an initial validation of the model. By collecting historical data on systems engineering projects from aerospace & defense companies that involved reuse and assessing the revised model's ability to estimate projects with reuse, the improvement to the estimation power of the COSYSMO model could be ascertained. To accomplish this, systems engineering project data was solicited from industry affiliates supporting this research. After several months of solicitation and discussion with subject matter experts, it was concluded that very limited data collection opportunities existed. Because systems engineering reuse was an emerging concept, few organizations collected historical information on reuse along with systems engineering effort data. Several affiliate organizations expressed interest in collecting data on systems engineering reuse going forward, but such data would not be available to support the near-term development of the model. In light of the limited data collection opportunities, this research re-examined the existing COSYSMO calibration data set and conducted multiple Delphi surveys with subject matter experts. Although actual project data was unavailable to support a validation, the results of the Delphi surveys provided sufficient information to support an initial validation of the model. The results of the existing COSYSMO calibration data set analysis and Delphi survey are discussed below.

A. COSYSMO Calibration Data Set Analysis

The existing COSYSMO calibration data set is the collection of systems engineering projects that were used to calibrate and validate the COSYSMO tool when it was published in 2005.[3] Data on these projects were obtained from personnel familiar with each project by populating a COSYSMO data collection instrument, which included values for all COSYSMO inputs as well as actual effort expended on the systems engineering project. In addition to collecting data on the inputs for the COSYSMO model, the instrument also asked responders about the amount of reuse in the system being reported. At the time, it was not intended for COSYSMO to estimate reuse, but the idea was that it may in the future and the data collection opportunity should attempt to obtain as much information as possible. However, from a modeling perspective, "reuse" was thought of in a much more limited capacity. In the data collection instrument, responders were asked to report the percentage of the value for each size driver that was reused. For example, if a responder reported the project had 100 nominal requirements and that 20 of those 100 requirements where reused, they would report 20% of the nominal requirements were reused. Out of the 42 projects in the data set, 54% reported some amount of reuse in one of the four size drivers and 13% reported reuse in all of the size drivers. With over half of all the projects in the data set reporting some amount of reuse, the potential for COSYSMO to overestimate the effort for projects with reuse became even more apparent.

A significant limitation with this approach of reporting the percentage reused is the lack of consistent definitions and the inability to account for varying levels of reuse. As a result, the various percentages of reuse reported in the data set did not always correlate with expected decreases in total effort. In other words, a single reuse category ("reuse" vs. "no reuse") inconsistently accounted for reuse. Therefore, a single reuse category did not appear to be adequate at accounting for systems engineering reuse.

When COSYSMO was published in 2005, the model was capable (at a minimum) of estimating the effort of a systems engineering project within 30% of the actual, 50% of the time, or PRED(30)=50%. To test the hypothesis that a single reuse category is inadequate, an experimental version of the COSYSMO 2.0 operational equation was created. Instead of parameters for six reuse categories, the modified model had a parameter for only one reuse category. Weights for this single reuse category were varied parametrically (from 0.0 to 1.5) and the estimation power of the experimental model was compared with the estimation power of the COSYSMO model over the same set of projects. Across this range of weights, the modified model consistently performed worse than COSYSMO without reuse. The inability of the modified model to improve the estimation power of COSYSMO appears to support the conclusion that a single reuse category is inadequate. This result also demonstrates the need for multiple reuse categories as well as consistent definitions of the categories.

B. COSYSMO 2.0 Delphi Survey

In the absence of actual project data, expert opinion is the next available source of validation. Due to the lack of existing historical systems engineering reuse data and the limited near-term data collection opportunities, the initial validation of COSYSMO 2.0 utilized subject matter expert opinion through a Delphi survey and obtained weights for the six reuse categories. To derive these weights, two rounds of a Delphi survey were conducted at COSYSMO Workshops, one at the 2008 COCOMO Forum in Los Angeles, CA and one at the 2009 Practical Software and Systems Measurement (PSM) Conference in Orlando, FL. The COSYSMO Workshops were selected as the venues for the survey for two reasons. First, one of the goals of each event was guide cost model research. Second, both events had a reasonable number of attendees (10-20 experts), most with a moderate to advanced familiarity with COSYSMO as well as previous experience in supporting the development of the model. Participants in the survey included representatives from organizations such as The Aerospace Corporation, BAE Systems, Boeing, Lockheed Martin, Northrop Grumman, and Raytheon. The weight derivation exercise assessed which of the standard systems engineering activities, by life-cycle phase, existed for each category of reuse. Table 4 provides the template and results for this exercise.

ISO/IEC 15288-Based Life Cycle Phases		Reuse Categories																							
EIA 632-Reuse Activity Cross Walk		SE Activities For DESIGN FOR REUSE				SE Activities For NEW				SE Activities For MODIFIED				SE Activities For DELETED				SE Activities For ADOPTED				SE Activities For MANAGED			
		Conceptualize	Develop	Operate Test & Eval	Transition to Operation	Conceptualize	Develop	Operate Test & Eval	Transition to Operation	Conceptualize	Develop	Operate Test & Eval	Transition to Operation	Conceptualize	Develop	Operate Test & Eval	Transition to Operation	Conceptualize	Develop	Operate Test & Eval	Transition to Operation	Conceptualize	Develop	Operate Test & Eval	Transition to Operation
Acquisition and Supply	1. Product Supply	X	X	X	X	X	X	X	X															X	
	2. Product Acquisition	X	X	X	X	X	X	X	X															X	
	3. Supplier Performance	X	X	X	X	X	X	X	X															X	
Technical Management	4. Process Implementation Strategy	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
	5. Technical Effort Definition	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
	6. Schedule and Organization	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
	7. Technical Plans	XX	XX	XX	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
	8. Work Directives	XX	XX	XX	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
	9. Progress Against Plans and Schedules	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
	10. Progress Against Requirements	XX	XX	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
	11. Technical Reviews	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
	12. Outcomes Management	XX	XX	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
	13. Information Dissemination	XX	XX	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
System Design	14. Acquirer Requirements	X	X	X	X	X	X	X	X	X				X				X							
	15. Other Stakeholder Requirements	XX	XX	X	X	X	X	X	X	X				X				X							
	16. System Technical Requirements	XX	XX	X	X	X	X	X	X	X				X				X							
	17. Logical Solution Representations	XX	XX	X	X	X	X	X	X	X				X				X							
	18. Physical Solution Representations	XX	XX	X	X	X	X	X	X	X				X				X							
	19. Specified Requirements	XX	XX	X	X	X	X	X	X	X				X				X							
Product Realization	20. Implementation	XX	XX	X	X	X	X	X	X	X	X			X	X										
	21. Transition to Use	X	X	X	X	X	X	X	X	X	X			X	X										
Technical Evaluation	22. Effectiveness Analysis	XX	XX	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	23. Tradeoff Analysis	XX	XX	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	24. Risk Analysis	XX	XX	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	25. Requirements Statements Validation	XX	XX	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	26. Acquirer Requirements Validation	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	27. Other Stakeholder Requirements Validation	XX	XX	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	28. System Technical Requirements Validation	XX	XX	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	29. Logical Solution Representations Validation	XX	XX	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	30. Design Solution Verification	XX	XX	XX	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	31. End Product Verification	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	32. Enabling Product Readiness	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				
	33. End Products Validation	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				

Table 4. COSYSMO 2.0 Delphi Results

COSYSMO is based on the 33 standard systems engineering activities defined by ANSI/EIA 632 (listed vertically in the first column) and guided by the four life-cycle stages in ISO/IEC 15288 (listed horizontally in the first row).[10,11] The exercise involved a review and discussion of each of the six reuse categories for each of the 33 activities, by life-cycle phase, to determine whether or not a particular activity would exist for that category, in that phase. One expert would typically propose their assessment of a particular relationship, discussion among the experts would occur, and a consensus would be reached.[12] If an activity was needed, an "X" was placed in the cell. If an activity was not needed, the cell was left blank. With the Design for Reuse category, 2 X's ("XX") represented instances where additional resources (greater than those for New) would be required to make a product reusable. At the COCOMO Forum, values for five of the six reuse categories (New, Modified, Deleted, Adopted, and Managed) were obtained. Due to time constraints, values for the sixth category were unable to be collected during this round of the Delphi. At the PSM Conference, values for the sixth reuse category (Design for Reuse) were collected.

After completing the matrix in Table 4, the weight of each reuse category was obtained by effectively adding the number of X's in each category. Previous research identified the standard effort associated with each activity and each life cycle phase. By summing the activities with an "X" in each category, the percentage of effort for each category compared with the New category could be obtained following a "bottom up" approach.[13] Since New (R_N) was assumed to be the effort associated with completing systems engineering products without reuse, products rated as Modified, Deleted, Adopted, or Managed should have fewer activities (less effort) comparatively, while products Designed for Reuse require additional time-consuming activities (more effort). The calculated weights are presented in Fig. 2. Although the exercise was primarily

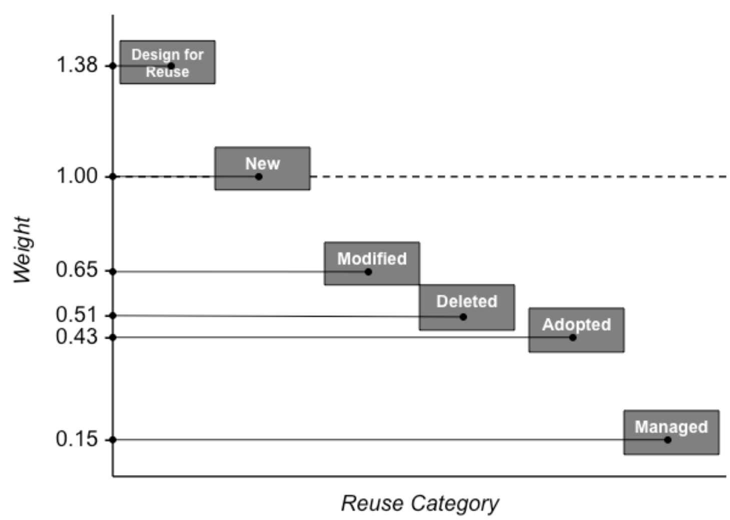

Figure 2. COSYSMO 2.0 Reuse Category Weights

focused on the effect of reuse on requirements, the results are believed to be applicable all COSYSMO size drivers and future research will further explore this applicability.

Overall, the outcomes of the reuse category weighting exercise aligned closely with the expected results. The Design for Reuse category was determined to require a significant upfront investment, 38% beyond the cost of building a New product, in order to make a product reusable and realize greater reuse benefits in the future. The Managed category was determined to offer substantial reuse savings, although few products may actually fit within its definition. One limitation of the exercise results is that the weights in Fig. 2 are point estimates, which represent the nominal reuse case, but not necessarily all reuse cases; some systems may have certain characteristics that influence the weights of the reuse categories beyond the nominal case. Future research will examine the possible ranges of these weights.

C. Example COSYSMO 2.0 Estimate

This example assumes a typical case in which a customer provides a system specification and requests an estimate of systems engineering effort from a contractor.

A customer has asked a contractor to estimate the amount of systems engineering effort that should be expected on an upcoming project. The customer provides the contractor with a system specification that contains 100 requirements (some reused), 10 interfaces (some reused), 7 system-specific algorithms, and 4 operational scenarios. Based on previous experiences with similar systems, the contractor has a high level of understanding of the requirements and there is a low level of technology risk.

Upon reviewing the provided data, the contractor determines there is sufficient data to utilize COSYSMO 2.0. After further review of the system specification, it is determined that the 100 requirements can be decomposed into 300 requirements at the systems engineering level. Through additional discussions with the customer, it is determined that of the 300 requirements at the systems engineering level; 100 are easy, 150 are nominal, and 50 are difficult. Following discussions with other experts at the contractor, it is determined that of the 100 easy requirements, 75 new and 25 are modified; of the 150 nominal requirements, 90 are new and 60 are adopted; and of the 50 difficult requirements, 20 are new and 30 are adopted. A similar evaluation process is followed for the interfaces and system-specific algorithms. While reviewing the 10 interfaces, it is determined that 2 of the interfaces will be reused in the future and they should be designed for reuse. In order to make the interface reusable and generate a reuse benefit for future systems, an upfront investment is made during this system development. Out of the 10 interfaces, 8 are nominal and new, and 2 are difficult and will be designed for reuse. All of the 7 system-specific algorithms as well as the 4 operational scenarios are nominal and new.

These values are inputted into the COSYSMO 2.0 size drivers and reuse extension, and provide an initial person-month estimate for the systems engineering activities based solely on the systems engineering size parameters. Additional information about the system and the capabilities of the systems engineering team can further adjust this estimate. Since it was determined that the contractor has a high level of understanding of the requirements and a low level of technology risk exists for this system, the requirements understanding and technology risk cost drivers can be adjusted to high and low, respectively. These adjustments to the cost drivers further align the estimate with the expected capabilities of the systems engineering team; however, this example includes cost driver inputs only as a means of illustrating the additional parameters that are available for the model user to best characterize the system of interest. Detailed discussions on the cost drivers were outside the scope of this paper.[3]

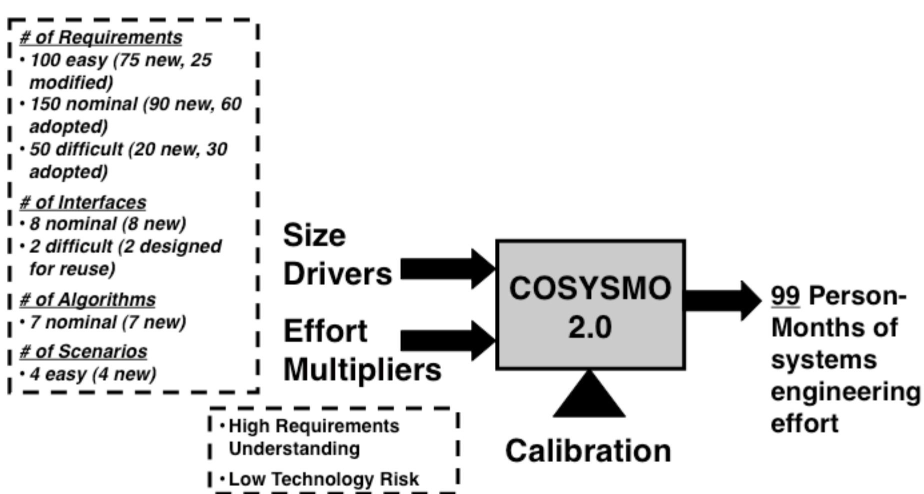

Figure 3. Example COSYSMO 2.0 estimate with reuse

Using the information from the system specification and supplementing it with customer and expert discussions, the COSYSMO 2.0 model inputs were populated and an estimate was generated. Based on the characteristics of this system and systems engineering team, COSYSMO 2.0 estimated 99.0 person-months to complete the systems engineering activities. An overview of the inputs and resulting estimate is provided in Figure 3. Comparatively, if COSYSMO (which does not account for the effect of reuse) was used instead of COSYSMO 2.0, the estimate would have been 129.1 person-months, an overestimate of 30.4%.

V. Conclusion

This paper presented the results of recent research supporting the definition and initial validation of the COSYSMO 2.0 model. Through participation at multiple workshops, discussions with dozens of subject matter experts, and presentations to COSYSMO supporters, industry consensus was obtained on the modeling methodology for COSYSMO 2.0. Part of this joint industry position included the establishment of industry-validated definitions for the six categories of systems engineering reuse. Due to the lack of historical data on systems engineering reuse and limited near-term data collection opportunities, the derivation of weights for each of the six reuse categories in COSYSMO 2.0 was determined with expert opinion through a Delphi survey. With these weights, an initial validation of the COSYSMO 2.0 model was finalized and an example estimate was provided.

Future COSYSMO 2.0 research will include further refinement to the estimation power of the model through calibration with historical data, and extrapolation of the reuse weights to test their robustness across the remaining size drivers and difference application domains. In addition, future research will include the development of operational guidance for the model and elaboration of the reuse process through the documentation of a reuse framework.[14]

Acknowledgments

The authors wish to thank The Aerospace Corporation for their support of this research, the USC Center for Systems and Software Engineering Corporate Affiliates and the MIT Lean Advancement Initiative Consortium Members, the attendees of the 2009 Practical Software and Systems Measurement Conference, and the participants of the 2008/2009 COSYSMO Workshops. In particular, Dr. Gan Wang of BAE Systems, Garry Roedler of Lockheed Martin and Dr. Barry Boehm of USC helped make this research possible.

References

[1]United States Government Accountability Office, “Best Practices for Estimating and Managing Program Costs,” Exposure Draft, GAO-07-1134SP, 2007.

[2]Valerdi, R., Wheaton, M. J., Fortune, J., “Systems Engineering Cost Estimation for Space Systems,” AIAA Space 2007, Long Beach, CA, September 2007.

[3]Valerdi, R., "The Constructive Systems Engineering Cost Model (COSYSMO)," PhD Dissertation, University of Southern California, 2005.

[4]Fortune, J., Valerdi, R., Boehm, B. W. and Settles, F. S., “Estimation Methods for Systems Engineering Reuse,” 7th Conference on Systems Engineering Research, Loughborough, U.K, April 2009.

[5]Valerdi, R., Gaffney, J., Roedler, G., Rieff, J., “Extensions of COSYSMO to Represent Reuse,” 21st Forum on COCOMO and Software Cost Modeling, Herndon, VA, October 2006.

[6] Rieff, J., Gaffney, J. and Roedler, G., “2007: The Breakout Year for COSYSMO,” PSM Users Group Conference, Golden CO, June 2007.

[7] Wang, G., “COSYSMO Extension: Reuse,” 2007 COCOMO Forum, COSYSMO Working Group, Los Angeles, CA, October 2007.

[8]Wang, G., Valerdi, R., Ankrum, A., Millar, C. and Roedler, G., “COSYSMO Reuse Extension,” 18th INCOSE Symposium, Utrecht, The Netherlands, June 2008.

[9]Boehm, B. W., Valerdi, R., Lane, J. and Brown, A. W., “COCOMO Suite Methodology and Evolution,” CrossTalk - The Journal of Defense Software Engineering, 18(4), 20-25, 2005.

[10]ANSI/EIA, “ANSI/EIA-632-1988 Processes for Engineering a System,” 1999.

[11]ISO/IEC, “ISO/IEC 15288:2002(E) Systems Engineering - System Life Cycle Processes,” 2002.

[12] Boehm, B. W., “Software Engineering Economics,” Prentice Hall, 1981.

[13]Fortune, J., “COSYSMO 2.0 Reuse Workshop Out-Brief,” Practical Systems and Software Measurement Conference, Orlando, FL, June 2009.

[14]Fortune, J. and Valerdi, R., “Considerations for Successful Reuse in Systems Engineering,” AIAA Space 2008, San Diego, CA, September 2008.