System-of-Systems Cost Estimation: Analysis of Lead System Integrator Engineering Activities

Jo Ann Lane, University of Southern California, USA; E-mail: jolane@usc.edu

Dr. Barry Boehm, University of Southern California, USA; E-mail: boehm@cse.usc.edu

Abstract

As organizations strive to expand system capabilities through the development of system-of-systems (SoS) architectures, they want to know "how much effort" and "how long" to implement the SoS. In order to answer these questions, it is important to first understand the types of activities performed in SoS architecture development and integration and how these vary across different SoS implementations. This paper provides results of research conducted to determine types of SoS Lead System Integrator (LSI) activities and how these differ from the more traditional system engineering activities described in Electronic Industries Alliance (EIA) 632 ("Processes for Engineering a System"). This research further analyzed effort and schedule issues on "very large" SoS programs to more clearly identify and profile the types of activities performed by the typical LSI and to determine organizational characteristics that significantly impact overall success and productivity of the LSI effort. The results of this effort have been captured in a reduced-parameter version of the Constructive SoS Integration Cost Model (COSOSIMO) that estimates LSI SoS Engineering (SoSE) effort.

Keywords: *System of Systems, System of Systems Engineering, Lead System Integrator, Cost Model.*

INTRODUCTION

As organizations strive to expand system capabilities through the development of system-of-systems (SoS) architectures, they want to know "how much effort" and "how long" to implement the SoS. Efforts are currently underway at the University of Southern California (USC) Center for Software Engineering (CSE) to develop a cost model to estimate the effort associated with SoS Lead System Integrator (LSI) activities. The research described in this paper is in support of the development of this cost model, the Constructive SoS Integration Cost Model (COSOSIMO). Research conducted to date in this area has focused more on technical characteristics of the SoS. However, feedback from USC CSE industry affiliates indicates that the extreme complexity typically associated with SoS architectures and political issues between participating organizations have a major impact on the LSI effort. This is also supported by surveys of system acquisition managers (Blanchette, 2005) and studies of failed programs (Pressman and Wildavsky, 1973). The focus of this current research is to further investigate effort and schedule issues on “very large” SoS programs and to determine key activities in the development of SoSs and organizational characteristics that significantly impact overall success and productivity of the program.

This paper first describes the context for the COSOSIMO cost model, then presents a conceptual view of the cost model that has been developed using expert judgment, describes the methodology being used to develop the model, and summarizes conclusions reached to date.

COSOSIMO CONTEXT

We are seeing a growing trend in industry and the government agencies to “quickly” incorporate new technologies and expand the capabilities of legacy systems by integrating them with other legacy systems, Commercial-Off-the-Shelf (COTS) products, and new systems into a system of systems, generally with the intent to share information from related systems and to create new, emergent capabilities that are not possible with the existing stove-piped systems.   With this development approach, we see new activities being performed to define the new architecture, identify sources to either supply or develop the required components, and then to integrate and test these high level components. Along with this “system-of-systems” development approach, we have seen a new role in the development process evolve to perform these activities: that of the LSI. A recent Air Force study (United States Air Force Scientific Advisory Board, 2005) clearly states that the SoS Engineering (SoSE) effort and focus related to LSI activities is considerably different from the more traditional system development projects. According to this report, key areas where LSI activities are more complex or different than traditional systems engineering are the system architecting, especially in the areas of system interoperability and system “ilities”; acquisition and management; and anticipation of needs.

Key to developing a cost model such as COSOSIMO is understanding what a “system-of-systems” is. Early literature research (Jamshidi, 2005) showed that the term “system-of-systems” can mean many things across different organizations. For the purposes of the COSOSIMO cost model development, the research team has focused on the SoS definitions provided in (Maier, 1999) and (Sage and Cuppan, 2001): an evolutionary net-centric architecture that allows geographically distributed component systems to exchange information and perform tasks within the framework that they are not capable of performing on their own outside of the framework. This is often referred to as “emergent behaviors”. Key issues in developing an SoS are the security of information shared between the various component systems, how to get the right information to the right destinations efficiently without overwhelming users with unnecessary or obsolete information, and how to maintain dynamic networks so that component system “nodes” can enter and leave the SoS.

Today, there are fairly mature tools to support the estimation of the effort and schedule associated with the lower-level SoS component systems (Boehm et al, 2005). However, none of these models supports the estimation of LSI SoSE activities. COSOSIMO, shown in Figure 1, is a parametric model currently under development to compute just this effort. The goal is to support activities for estimating the LSI effort in a way that allows users to develop initial estimates and then conduct tradeoffs based on architecture and development process alternatives.

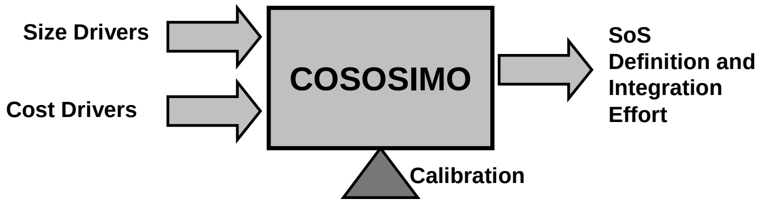


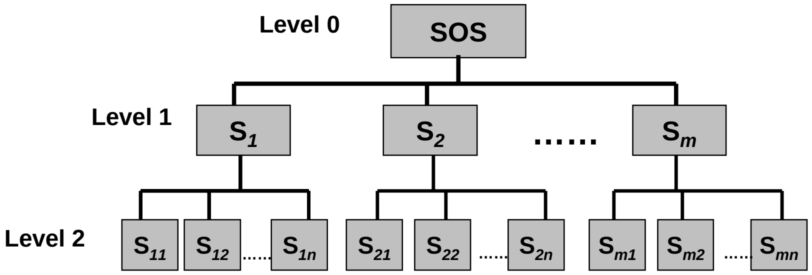
Figure 1. COSOSIMO Model Structure.

Recent LSI research conducted by reviewing LSI statements of work identifies the following typical LSI activities:

- Concurrent engineering of requirements, architecture, and plans

- Identification and evaluation of technologies to be integrated
- Source selection of vendors and suppliers
- Management and coordination of supplier activities
- Validation and feasibility assessment of SoS architecture
- Continual integration and test of SoS-level capabilities
- SoS-level implementation planning, preparation, and execution
- On-going change management at the SoS level and across the SoS-related integrated product teams to support the evolution of requirements, interfaces and technology.

With the addition of this new cost model to the Constructive Cost Model (COCOMO) suite of cost models, one can easily develop more comprehensive estimates for the total SoS development, as shown in Figure 2.



Activity	Levels	Cost Model
SoS Lead System Integrator Effort (SoS scoping, planning, requirements, architecting; source selection; teambuilding, re-architecting, feasibility assurance with selected suppliers; incremental acquisition management; SoS integration and test; transition planning, preparation, and execution; and continuous change, risk, and opportunity management)	Level 0, and other levels if lower level systems components are also SoSs	COSOSIMO
Development of SoS Software-Intensive Infrastructure and Integration Tools	Level 0	COCOMO II
System Engineering for SoS Components	Levels 1-n	COSYSMO
Software Development for Software-Intensive Components	Levels 1-n	COCOMO II
COTS Assessment and Integration for COTS-based Components	Levels 1-n	COCOTS

Figure 2. Architecture-based SoS Cost Estimation.

LSI EFFORT ESTIMATION APPROACH

As mentioned above, key to an LSI effort estimation model is having a clear understanding of the SoSE activities performed by the organization as well as which activities require the most effort. In addition, it is important to understand how these SoSE activities differ from the more traditional systems engineering activities. Analysis presented in (Lane 2005) describes how the typical LSI SoSE activities differ from the more traditional system engineering activities identified in EIA 632 (Electronic Industries Alliance, 1999) and the Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI) (Software Engineering Institute, 2001). Subsequently, Delphi surveys conducted with USC CSE industry affiliates have identified key size drivers and cost drivers for LSI effort and are shown in Table 1.

Table 1. COSOSIMO Cost Model Parameters

Size Drivers	Cost Drivers
• # SoS-related requirements • # SoS interface protocols • # independent component system organizations • # SoS scenarios • # unique component systems	• Requirements understanding • Architecture maturity • Level of service requirements • Stakeholder team cohesion • SoS team capability • Maturity of LSI processes • Tool support • Cost/schedule compatibility • SoS Risk Resolution • Component system maturity and stability • Component system readiness

Because there are concerns about the availability of effort data from a sufficient number of SoS programs to support model calibration and validation, current efforts are focussing on defining a “reduced parameter set” cost model or ways to estimate parts of the LSI effort using fewer, but more specific, parameters. The following paragraphs present the results of this recent research.

Further observations of LSI organizations indicate that the LSI activities can be grouped into three areas: 1) planning, requirements management, and architecting (PRA), 2) source selection and supplier oversight (SS), and 3) SoS integration and testing (I&T). There are typically different parts of the LSI organization that are responsible for these three areas. Figure 3 illustrates, conceptually, how effort for these three areas is distributed across the SoS development life cycle phases of inception, elaboration, construction, and transition for a given increment or evolution of SoS development.

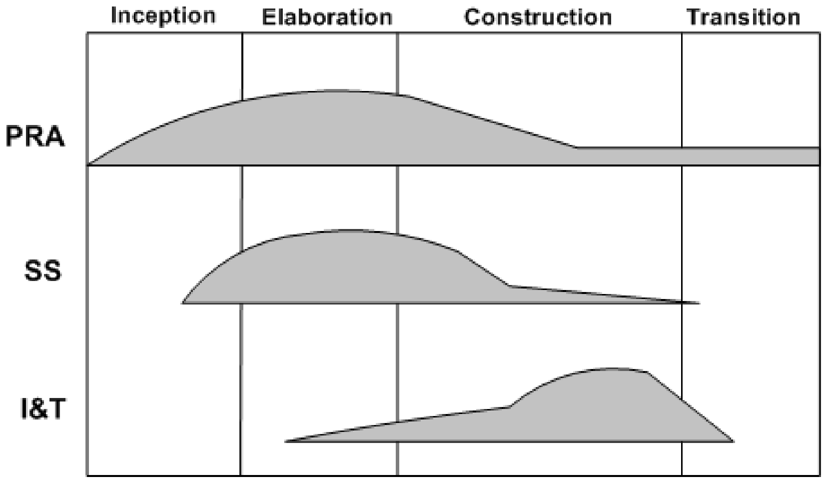


Figure 3. Conceptual LSI Effort Profile.

Planning, requirements, and architecting begin early in life cycle. As the requirements are refined and the SoS architecture is defined and matured, source selection activities can begin to identify component system vendors and to issue contracts to incorporate the necessary SoS-enabling capabilities. With a mature SoS architecture and the identification of a set of component systems for the current increment, the integration team can begin the integration and test planning activities. Once an area ramps up, it continues through the transition phase at some nominal level to ensure as smooth a transition as possible and to capture lessons learned to support activities and plans for the next increment. (Boehm and Lane, 2006) describes how some of these activities directly support the current plan-driven SoS development effort while others are more agile, forward looking, trying to anticipate and resolve problems before they become huge impacts. The goal is to stabilize development for the current increment while deferring as much change as possible to future increments. For example, the planning/requirements/architecture group continues to manage the requirements change traffic that seems to be so common in these large systems, only applying those changes to the current increment that are absolutely necessary, and deferring the rest to future increments. The architecture team also monitors current increment activities in order to make necessary adjustments to the architecture to handle cross-cutting technology issues that arise during the component system supplier construction activities. Likewise, the supplier oversight group continues to monitor the suppliers for risks, cost, and, schedule issues that arise out of SoS conflicts with the component system stakeholder needs and desires. As the effort ramps down in the transition phase, efforts are typically ramping up for the next increment or evolution.

By decomposing the COSOSIMO cost model into three components that correspond to the three primary areas of LSI SoSE effort, the parameter set for each COSOSIMO component can be reduced from the full set and the applicable cost drivers made more specific to the target area. Table 2 shows the resulting set of size and cost drivers for each of the three primary areas. This approach allows the model developers to calibrate and validate the model components with fewer parameters and data sets. It also allows the collection of data sets from organizations that are only responsible for a part of the LSI SoSE activities. Finally, this approach to LSI SoSE effort estimation allows the cost model to provide estimates for the three areas, as well as a total estimate—a key request from USC CSE industry affiliates supporting this research effort.

Table 2. COSOSIMO Parameters by SoSE Area.

COSOSIMO Component	Associated Size Drivers	Associated Cost Drivers
PRA	• # SoS-related requirements • # SoS interface protocols	• Requirements understanding • Level of service requirements • Stakeholder team cohesion • SoS PRA capability • Maturity of LSI PRA processes • PRA tool support • Cost/schedule compatibility with PRA processes • SoS PRA risk resolution
SS	• # independent component system organizations	• Requirements understanding • Architecture maturity • Level of service requirements • SoS SS capability • Maturity of LSI SS processes • SS tool support • Cost/schedule compatibility with SS activities • SoS SS risk resolution
I&T	• # SoS interface protocols • # SoS scenarios • # unique component systems	• Requirements understanding • Architecture maturity • Level of service requirements • SoS I&T capability • Maturity of LSI I&T processes • I&T tool support • Cost/schedule compatibility with I&T activities • SoS I&T risk resolution • Component system maturity and stability • Component system readiness

Detailed definitions and proposed ratings for these parameters may be found in (Lane, 2006). The following provides a brief description of each of the COSOSIMO parameters. Note that several of the COSOSIMO parameters are similar to those defined for the Constructive Systems Engineering Cost Model (COSYSMO) and are identified in the descriptions below.

COSOSIMO Size Drivers

Number of SoS-Related Requirements[i]: This driver represents the number of requirements for the SoS of interest at the SoS level. Requirements may be functional, performance, feature, or service-oriented in nature depending on the methodology used for specification. They may also be defined by the customer or contractor. SoS requirements can typically be quantified by counting the number of applicable *shalls*, *wills*, *shoulds*, and *mays* in the SoS or marketing specification. Note: some work may be required to decompose requirements to a consistent level so that they may be counted accurately for the appropriate SoS-of-interest.

Number of SoS Interface Protocols: The number of distinct net-centric interface protocols to be provided/supported by the SoS framework. Note: This does NOT include interfaces internal to the SoS component systems, but it does include interfaces external to the SoS and between the SoS component systems. Also note that this is not a count of total interfaces (in many SoSs, the total number of interfaces may be very dynamic as component systems come and go in the SoS environment—in addition, there may be multiple instances of a given type of component system), but rather a count of distinct protocols at the SoS level.

Number of Independent Component System Organizations: The number organizations managed by the LSI that are providing SoS component systems.

Number of Operational Scenarios[i]: This driver represents the number of operational scenarios that an SoS must satisfy. Such scenarios include both the nominal stimulus-response thread plus all of the off-nominal threads resulting from bad or missing data, unavailable processes, network connections, or other exception-handling cases. The number of scenarios can typically be quantified by counting the number of SoS states, modes, and configurations defined in the SoS concept of operations or by counting the number of “sea-level” use cases (Cockburn, 2001), including off-nominal extensions, developed as part of the operational architecture.

Number of Unique Component Systems: The number of types of component systems that are planned to operate within the SoS framework. If there are multiple versions of a given type that have different interfaces, then the different versions should also be included in the count of component systems.

COSOSIMO Cost Drivers

Requirements Understanding[i]: This cost driver rates the level of understanding of the SoS requirements by all of the affected organizations. For the PRA sub-model, it includes the PRA team as well as the SoS customers and sponsors, SoS PRA team members, component system owners, users, etc. For the SS sub-model, it is the understanding level between the LSI and the component system suppliers/vendors. For the I&T sub-model, it is the level of understanding between all of the SoS stakeholders with emphasis on the SoS I&T team members.

Level of Service Requirements[i]: This cost driver rates the difficulty and criticality of satisfying the ensemble of level of service requirements or Key Performance Parameters (KPPs), such as security, safety, transaction speed, communication latency, interoperability, flexibility/adaptability, and reliability. This parameter should be evaluated with respect to the scope of the sub-model to which it pertains.

Team Cohesion[i]: Represents a multi-attribute parameter which includes leadership, shared vision, diversity of stakeholders, approval cycles, group dynamics, Integrated Product Team (IPT) framework, team dynamics, trust, and amount of change in responsibilities. It further represents the heterogeneity in stakeholder community of the end users, customers, implementers, and development team. For each sub-model, this parameter should be evaluated with respect to the appropriate LSI team (e.g., PRA, SS, or I&T).

Team Capability: Represents the anticipated level of team cooperation and cohesion, personnel capability and continuity, as well as LSI personnel experience with the relevant domains, applications, language, and tools. For each sub-model, this parameter should be evaluated with respect to the appropriate LSI team (e.g., PRA, SS, or I&T).

Process Maturity: A parameter that rates the maturity level and completeness of the LSI's processes and plans. For each sub-model, this parameter should be evaluated with respect to the appropriate LSI team processes (e.g., PRA, SS, or I&T).

Tool support[i]: Indicates the coverage, integration, and maturity of the tools in the SoS engineering and management environments. For each sub-model, this parameter should be evaluated with respect to the tool support available to appropriate LSI team (e.g., PRA, SS, or I&T).

Cost/Schedule Compatibility: The extent of business or political pressures to reduce the cost and schedule associated with the LSI's activities and processes. For each sub-model, this parameter should be evaluated with respect to the cost/schedule compatibility for appropriate LSI team activities (e.g., PRA, SS, or I&T).

Risk Resolution: A multi-attribute parameter that represents the number of major SoS/LSI risk items, the maturity of the associated risk management and mitigation plan, compatibility of schedules and budgets, expert availability, tool support, and level of uncertainty in the risk areas. For each sub-model, this parameter should be evaluated with respect to the risk resolution activities for the associated LSI team (e.g., PRA, SS, or I&T).

Architecture Maturity: A parameter that represents the level of maturity of the SoS architecture. It includes the level of detail of the interface protocols and the level of understanding of the performance of the protocols in the SoS framework. Two COSOSIMO sub-models use this parameter and it should be evaluated in each case with respect to the LSI activities covered by the sub-model of interest.

Component System Maturity and Stability: A multi-attribute parameter that indicates the maturity level of the component systems (number of new component systems versus number of component systems currently operational in other environments), overall compatibility of the component systems with each other and the SoS interface protocols, the number of major component system changes being implemented in parallel with the SoS framework changes, and the anticipated change in the component systems during SoS integration activities.

Component System Readiness: Indicates readiness of component systems for integration. User evaluates level of Verification and Validation (V&V) that has/will be performed prior to integration and the level of subsystem integration activities that will be performed prior to integration into the SoS integration lab.

COSOSIMO COST MODEL DEVELOPMENT METHODOLOGY

The COSOSIMO cost model is being developed using the proven cost model development methodology developed over the last several years at the USC CSE. This methodology, described in (Boehm et al, 2002), is illustrated in Figure 4.

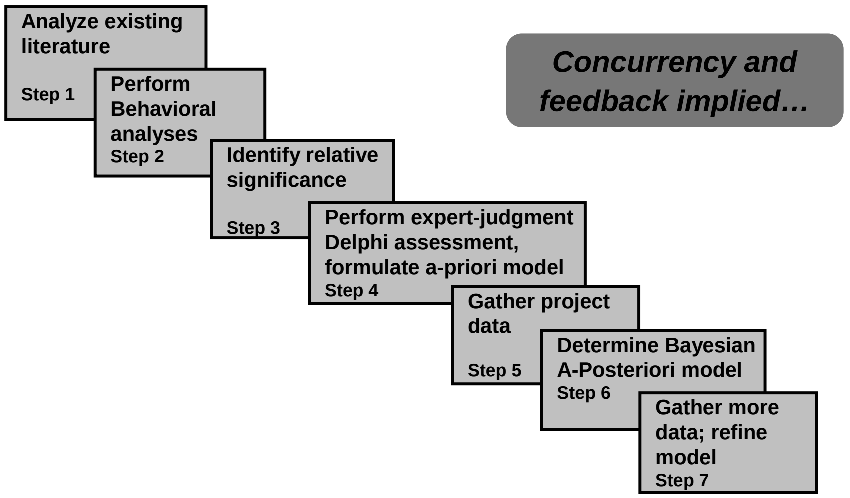


Figure 4. USC CSE Cost Model Development Methodology.

For COSOSIMO, the *literature review* has focused on the definitions of SoSs and SoSE; the role and scope of activities typically performed by LSIs; analysis of cost factors used in related software, systems engineering, and COTS integration cost models; as well as related system dynamics models that investigate candidate SoSE cost factors.

The *behavioral analyses* determine the potential range of values for the candidate cost drivers and the relative impact that each has on the overall effort associated with the relevant SOSE activities. For example, if the stakeholder team cohesion is very high, what is the impact on the PRA effort. Likewise, if the stakeholder team cohesion is very low, what is the resulting impact on PRA effort. The results of the behavioral analyses are then used to *develop a preliminary model form*.

The parameters include a set of one or more size drivers, a set of exponential scale factors, and a set of effort multipliers. Cost drivers that are related to economies/diseconomies of scale as size is increased are combined into an exponential factor. Other cost drivers that have a more linear behavior with respect to size drivers are combined into an effort multiplier.

Next, the model parameters, definitions, range of values, rating scales, and behaviors are reviewed with industry and research experts using *a wideband Delphi process*. The consensus of the experts is used to update the preliminary model. In addition to expert judgement, *actual effort data* is collected from successful projects covering the LSI activities of interest. A second model, based on actual data fitting, is then developed. Finally, the expert judgment and actual data *models are combined using Bayesian techniques*. In this process, more weight is given to expert judgement when actual data is not consistent or sparse and more weight is given to actual data when the data is fairly consistent and experts do not strongly agree.

Since technologies and engineering approaches are constantly evolving, it is important to *continue data collection and model analysis* and update the model when appropriate. Historically, this has led to parameters related to older technologies being dropped and new parameters added. In the case of COSOSIMO, it will be important to track the evolution of SoS architectures and integration approaches and the development of convergence protocols.

For COSOSIMO, each of the sub-models will go through this development process. Once the sub-models are calibrated and validated, they may be combined to estimate the total LSI effort for a proposed SoS development program. To date, several expert judgment surveys have been conducted and actual data collection is in process.

CONCLUSIONS

LSI organizations are realizing that if more traditional processes are used to architect and integrate SoSs, it will take too long and too much effort to find optimal solutions and build them. Preliminary analysis of LSI activities shows that while many of the LSI activities are similar to those described in EIA 632 and the SEI's CMMI, LSIs are identifying ways to combine agile processes with traditional processes to increase concurrency, reduce risk, and further compress overall schedules. In addition, effort profiles for the key LSI activities (the up-front effort associated with SoS abstraction, architecting, source selection, systems acquisition, and supplier and vendor oversight during development, as well as the effort associated with the later activities of integration, test, and change management) show that the percentage of time spent on key activities differs considerably from the more traditional system engineering efforts. By capturing the effects of these differences in organizational structure and system engineering processes in a reduced parameter version of COSOSIMO, management will have a tool that will better predict LSI SoSE effort and to conduct "what if" comparisons of different development strategies.

REFERENCES

Blanchette, S. (2005). U.S. Army acquisition – The Program Executive Officer perspective, Special Report CMU/SEI-2005-SR-002. Pittsburg, PA: Software Engineering Institute.

Boehm, B., et al. (2000). Software cost estimation with COCOMO II. Upper Saddle River, NJ: Prentice Hall.

Boehm, et al. (2005). COCOMO suite methodology and evolution. CrossTalk, 18(4), 20-25.

Boehm, B., and Lane, J. (2006). 21st century processes for acquiring 21st century systems of systems. CrossTalk, 19(5), 4-9.

Cockburn, A.(2001). Writing effective use cases. Boston, MA: Addison-Wesley.

Electronic Industries Alliance (1999). EIA Standard 632: Processes for engineering a system.

Jamshidi, M. (2005). System-of-systems engineering - a definition; IEEE System, Man, and Cybernetics (SMC) Conference. Retrieved January 29, 2005 from the World Wide Web: http://ieeesmc2005.unm.edu/SoSE_Defn.htm

Lane, J. (2005). System of systems lead system integrators: Where do they spend their time and what makes them more/less efficient; USC-CSE-TR-2005-508. Los Angeles, CA: University of Southern California Center for Software Engineering.

Lane, J (2006). COSOSIMO Parameter Definitions, USC-CSE-TR-2006-606. Los Angeles, CA: University of Southern California Center for Software Engineering.

Maier, M. (1998). Architecting principles for systems-of-systems. Systems Engineering, 1(4), 267-284.

Pressman, J. and Wildavsky, A. (1973). Implementation: How great expectations in Washington are dashed in Oakland; Oakland, CA: University of California Press.

Sage, A., and Cuppan, C. (2001). On the systems engineering and management of systems of systems and federations of systems. Information, Knowledge, and Systems Management 2, 325-345.

Software Engineering Institute (2001). Capability maturity model integration (CMMI); CMU/SEI-2002-TR-001. Pittsburg, PA: Software Engineering Institute.

United States Air Force Scientific Advisory Board (2005). Report on system-of-systems engineering for Air Force capability development; Public Release SAB-TR-05-04. Washington, DC: HQ USAF/SB.

Valerdi, R (2005). The constructive systems engineering cost model (COSYSMO); Unpublished PhD Dissertation. Los Angeles, CA: University of Southern California.

---

[i] Adapted to SoS environment from COSYSMO (Valerdi, 2005).