Advancing an Ontology for Systems Engineering to Allow Consistent Measurement

Eric C. Honour
Systems Engineering & Evaluation Centre
University of South Australia
Mawson Lakes, SA
eric.honour@postgrads.unisa.edu.au
+1 (850) 479-1985 (Pensacola, FL)

Ricardo Valerdi, Ph.D.
MIT Lean Aerospace Initiative
77 Vassar Street, Building 41-205
Cambridge, MA 02139
rvalerdi@mit.edu
+1 (617) 253-8583

Abstract. Past analysis has shown that there is a quantifiable correlation between the amount, types and quality of systems engineering efforts used during a program and the success of the program. For any given program, an amount, type and quality of systems engineering effort can be selected from the quantified correlations. The optimal nature of these selections, however, has not yet been explored. An ongoing project, Systems Engineering Return on Investment (SE-ROI), aims to quantify the correlations by gathering data on current and completed programs. It is the purpose of this paper to advance an ontology that can support useful quantification of the desired correlations. This ontology is based on a review of current systems engineering standards, historical systems engineering activities, and data gathered on the COSYSMO and Value of Systems Engineering projects. In this paper, the ontology is further explored to create broadly-based definitions of key terms such as "systems engineering effort," "amount of effort," "type of effort," "quality," "success," and "optimum." The SE-ROI project is continuing to convert the ontology into a methodology for measuring Return on Investment. This measurement will yield more specific relationships between systems engineering activities, such as requirements management effort, and the cost/schedule compliance of the program.[1]

Background

In prior work [Honour 2004], the authors have shown that the discipline of systems engineering (SE) has been recognized for 50 years as essential to the development of complex systems. They have also shown how SE is still treated primarily as heuristics learned by each practitioner during the personal experimentation of a career. The heuristics known by each differ, as shown by the fractured development of SE "standards" and SE certification. The standards discussed in this paper include ANSI/EIA-632, IEEE-1220, ISO-15288, Capability Maturity Model Integration (CMMI)[2], and MIL-STD-499C. These standards provide indication of good practices that are agreed by the authors of the standards, but largely do not provide proof. Some few research projects provide quantitative indications of the value of SE, but the information available lacks strong statistical evidence.

As a result of this heuristic understanding of the discipline, it has been nearly impossible to quantify the value of SE to programs. [Sheard 2000] Yet both practitioners and managers intuitively understand that value. They typically incorporate some SE practices in every

[1] This paper was published in the proceedings of the Conference on Systems Engineering Research, Los Angeles, CA, 2006.
[2] "Capability Maturity Model," "Capability Maturity Model Integration," and "CMMI" are registered trademarks of the Carnegie Mellon University Software Engineering Institute.

complex program. As an example, the US Department of Defense has mandated the use of a "robust SE approach," although with little definition of what this means. [Wynne 2004] The differences in understanding, however, just as typically result in disagreement over the level and formality of the practices to include. [Shenhar 1997] In response to the uncertainty in the heuristics, some efforts have been made to identify "leading indicators" that provide early tracking information as to the worth of a project's SE efforts. [Roedler 2005]

Recent work is beginning to quantify systems engineering. The COSYSMO project [Valerdi 2004] has created a constructive cost model for systems engineering based on both gathered heuristics and data from real programs. The result is an initial model that provides an indication of the manpower level of systems engineering that matches the best of the real programs. The "Value of Systems Engineering" project [Honour 2004] has correlated subjective submissions for SE effort with cost, schedule, and perceptive success of the programs. Both projects have encountered two difficulties: scarce systems engineering data and wide variance in definitions and perceptions of systems engineering.

SE-ROI Project

The SE Return on Investment (SE-ROI) project seeks to gather empirical information to understand how systems engineering methods relate to program success (cost, schedule, and technical terms). In particular, the project expects to achieve three practical results:

1. **Identification of good SE practices** that correlate with success under different conditions.
2. **Leading indicators** that can be used during a program to assess the program's expected future success and risks based on SE practices used.
3. **Statistical correlation of SE methods with program success**, to understand how much of each SE method is appropriate under what conditions.

To achieve these results, the project plans to obtain access to in-process and recently completed programs. Work in the "Value of Systems Engineering" project has gathered data that is limited by the subjectivity of the information, the memories of the participants, and the volunteer nature of participants. Reducing these limitations requires actual data from programs, obtained through a series of structured interviews with key individuals on each program.

The data required includes:

- Program characterization data such as program size, program type, development phases, bounding parameters, risk levels.
- Program success data such as cost/schedule compliance and technical quality measures.
- Systems engineering data such as hours expended on systems engineering tasks, quality of those tasks, specific nature of the methods and tools used,

The plan is that access to programs will be obtained through sponsorship within government and industry. As such, the project is approaching various funding and non-funding individuals who can provide access (or impetus for access) to specific programs. Some data should also be available through collaboration with the COSYSMO project at the University of Southern California.

An Ontology for Quantification

According to Merriam-Webster, ontology is a "branch of metaphysics concerned with the nature and relations of being." A more specific use of the word was proposed by [Studer 1998] as representing a shared conceptualization. One of the greatest difficulties in quantifying systems engineering is the lack of such a shared conceptualization on the field of study. [Kasser 1995] Yet the field of systems engineering obviously exists as evidenced by the numerous academic programs and employment opportunities available. Therefore, the first step in quantification is to discover a shared definition of the field in terms of the nature and relations of the activities that are commonly considered to be "systems engineering." It is the purpose of this paper to advance an ontology that can support useful quantification of the desired correlations. This paper adds further thoughts to [Honour 2005].

Defining the Field of Study. For the SE-ROI purpose, the field of "systems engineering" is taken in a broad sense that includes all efforts that apply science and technology ("engineering") to the development of interacting combinations of elements ("systems"). Such efforts are frequently characterized as having both technical and management portions because of the inter-disciplinary nature of system development teams. The breadth of skills necessary for good SE was studied well by [Frank 2000] and typically includes skills of technical domain expertise, technical planning/management, and leadership.

Need for an Ontology for Quantification. The desired results for the SE-ROI project include correlation of specific SE methods with the success of the program. Doing so requires obtaining data through interviews that has sufficient structure to support statistical analysis of the correlations. Interview data sheets are now being designed to obtain this data, but the structure of the sheets must reflect a generally agreed structure of systems engineering.

Methodology. To obtain such a shared conceptualization, the authors have examined information that indicates shared and differing views of systems engineering. In the COSYSMO project, data from multiple systems engineering organizations allows structuring a profile of the shared views. This data is reviewed in the first subsection below. In the second subsection below, the authors show a review of the current systems engineering standards to find what elements they have in common and in difference. Those elements on which the standards agree indicate the shared conceptualizations.

COSYSMO Systems Engineering Effort Profile

One structure was explored as a part of the COSYSMO project, based primarily on the ANSI/EIA-632 standard [Valerdi 2005]. While this standard is not the only SE standard (see the next section), the insight obtained is useful to understanding the ontology. The actual application of the ANSI/EIA standard was found to be different in each organization studied[3]. Before seeking to obtain data on systems engineering effort for the calibration of COSYSMO, the necessary life cycle phases of interest were defined through the use of a recently developed standard.

Life Cycle Phases. A definition of the system life cycle phases was needed to help define the model boundaries. Because the focus of COSYSMO is systems engineering, it employs some of the life cycle phases from ISO/IEC 15288 *Systems Engineering – System Life Cycle Processes* [ISO 2002]. These phases were slightly modified to reflect the influence of the aforementioned model, ANSI/EIA 632, and are shown in Figure 1.

Life cycle models vary according to the nature, purpose, use and prevailing circumstances of the system. Despite an infinite variety in system life cycle models, there is an essential set of characteristic life cycle phases that exist for use in the systems engineering domain. For example, the *Conceptualize* phase focuses on identifying stakeholder needs, exploring different

[3] BAE Systems, General Dynamics, Lockheed Martin, Northrop Grumman, Raytheon, SAIC

Figure 1. ISO 15288 Life Cycle Phases.

solution concepts, and proposing candidate solutions. The *Development* phase involves refining the system requirements, creating a solution description, and building a system. The *Operational Test & Evaluation* phase involves verifying/validating the system and performing the appropriate inspections before it is delivered to the user. The *Transition to Operation* phase involves the transition to utilization of the system to satisfy the users' needs. The scope of COSYSMO is limited to these four life cycle phases. The final two were included in the data collection effort but did not yield enough data to be useful in the model calibration. These phases are: *Operate, Maintain, or Enhance* which involves the actual operation and maintenance of the system required to sustain system capability, and *Replace or Dismantle* which involves the retirement, storage, or disposal of the system.

The typical distribution of systems engineering effort across the first four life cycle phases for the organizations studied was obtained and is shown in Table 1. It is important to note that the standard deviation for each of the phases is relatively high. This is quantitative evidence for the argument that SE is applied very differently across organizations.

Processes for Engineering a System. The ANSI/EIA 632 model provides a generic list of SE activities that may or may not be applicable to every situation, but was deemed useful in describing the scope of systems engineering for COSYSMO. Other types of systems engineering activities lists exist, such as the one developed by Raytheon Space & Airborne Systems [Ernstoff 1999]. Such lists provide, in much finer detail, the common activities that are likely to be performed by systems engineers in those organizations, but are generally not applicable outside of the companies or application domains in which they are created. The typical distribution of systems engineering effort across the fundamental process areas for the organizations studied was collected and is shown in Table 2.

Table 1. Systems Engineering Effort across ISO 15288 Life Cycle Phases.

Phase	Conceptualize	Develop	Operational Test & Eval	Transition to Operation
%Effort (STDEV)	23 (12)	36 (16)	27 (13)	14 (9)

Table 2. Effort Distribution across ANSI/EIA 632 Fundamental Processes.

ANSI/EIA 632 Fundamental Process	Average	Standard Deviation
Acquisition & Supply	7%	3.5
Technical Management	17%	4.5
System Design	30%	6.1
Product Realization	15%	8.7
Technical Evaluation	31%	8.7

The results in Tables 1 and 2 can be combined to produce a detailed allocation of processes across phases as shown in Table 3. This information can help produce staffing charts that are helpful in determining the typical distribution of systems engineering effort for aerospace programs. Each program will have its own unique staffing profile based on the project characteristics and system complexity. Moreover, some organizations may not be responsible for the systems engineering involved with all four phases being shown here. In these cases, these organizations must interpolate the data provided in Tables 1, 2 and 3.

Table 3. Effort Distribution of ANSI/EIA 632 Fundamental Processes across ISO 15288 Phases

	Conceptualize	Develop	Operational Test & Eval.	Transition to Operation	(checksum)
Acquisition and Supply	28 (12.3)	51 (18.6)	13 (11.3)	8 (5.0)	100
Technical Management	22 (10.0)	38 (9.9)	25 (7.4)	15 (6.4)	100
System Design	34 (12.4)	40 (19.4)	17 (9.6)	9 (6.2)	100
Product Realization	13 (14.1)	30 (24.3)	32 (16.0)	25 (20.4)	100
Technical Evaluation	18 (11.4)	27 (11.0)	40 (17.7)	15 (8.5)	100

The information in Table 3 can be graphically represented as a staffing profile chart as illustrated in Figure 2. This view is compatible with many of the cost estimation models used by project managers.

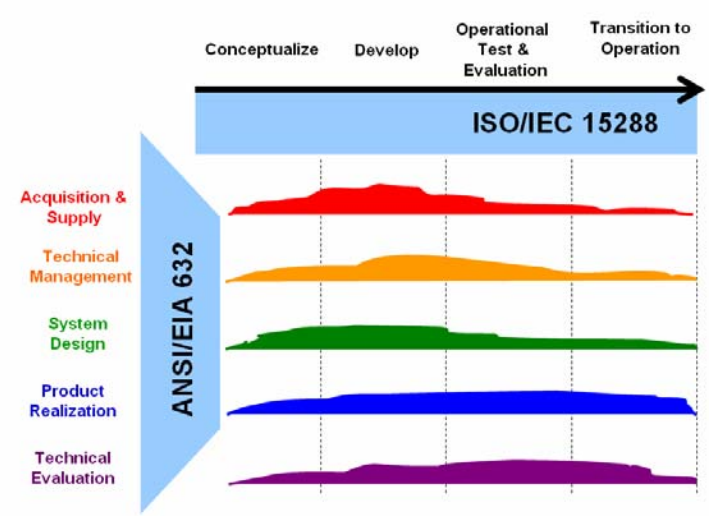

Figure 2. Systems Engineering Effort Profile.

Review of Systems Engineering Standards.

The COSYSMO work was based on only two standards. The lack of agreement on the field of study is reflected in the results in Tables 1 and 2; and reaffirmed by the current state of systems engineering standards. There are at least five “standards” for the field that are in wide use, each one used by different organizations, each with their own proponents and purposes. Table 4 lists these standards. Others also exist that are used in specialized domains, such as

Table 4. Notable Systems Engineering Standards

Standard (year)	Title
ANSI/EIA-632 (1999)	Processes for Engineering a System
IEEE-1220 (1998)	Application and Management of the Systems Engineering Process
ISO/IEC 15288 (2002)	Systems Engineering – System Life Cycle Processes
CMMI® (2002)	Capability Maturity Model® Integration℠ (CMMI®)
MIL-STD-499C (2005)	Systems Engineering

ECSS-E-10A used by the European Space Agency. It is beyond the scope of this paper to attempt to characterize these standards. Instead, we simply seek to correlate the information in them to discover the shared conceptualizations that represent the current ontology of SE.

Table 5 shows and compares the content of the current standards, in the context of an ontology that describes the relationship of various categories of effort that are widely viewed as "systems engineering." Eight primary categories of effort appear to be shared by most of the standards. As can be seen, these categories appear in each standard using somewhat different language and widely different descriptions. The eight categories are:

Mission/Purpose Definition. The starting point for the creation of a new system, or the modification of an existing system, is to define the mission or purpose of the new/changed system. This mission is typically described in the language of the system users rather than in technical language (i.e. the range of an airplane rather than the length, drag, wingspan, tank capacity, fuel rate, etc.) In the contracted systems environment (as opposed to the product systems environment), this task is often performed by a contracting agency before involving a systems development company. Because creation of most of the systems engineering standards has been driven by the contracted systems environment, several of the standards do not include this activity as part of systems engineering. Yet even in this environment, the activity is widely recognized as one performed by the "systems engineers" within the contracting agencies.

Requirements Engineering. A long-recognized core discipline of systems engineering has been the creation and management of requirements, formal technical statements that define the capabilities, characteristics, or quality factors of a system. Generally referred to as "requirements management" or "requirements engineering," this discipline may include efforts to define, analyze, validate, and manage the requirements. Because these efforts are so widely recognized, they appear in every standard.

System Architecting. The design aspect of systems engineering is to define the system in terms of its component elements and their relationships. This category of effort has come to be known as "architecting," following the practice of civil engineering in which the structure, aesthetics and relationship of a building are defined before doing the detailed engineering work to design the components. In systems engineering, architecting takes the form of diagrams that depict the high-level concept of the system in its environment, the components of the system, and the relation of the components to each other and to the environment. Creation of the system architecture (sometimes called "system design") is usually described as a process of generation and evaluation of alternatives. As a part of architecting, systems engineers define the components in terms of "allocated requirements" through a process of defining lower-level requirements from the system requirements.

Table 5. Systems Engineering Effort Categories Evident in the Standards

SE Categories	ANSI/EIA-632	IEEE-1220	ISO-15288	CMMI	MIL-STD-499C
Mission/purpose definition	Not included in scope	▪ Define customer expectations (Req Anlys)	▪ Stakeholder needs definition	▪ Develop customer requirements (Req Devlp)	Not included in scope
Requirements engineering	System Design ▪ Requirements definition	▪ Requirements analysis ▪ Track requirements and design changes	▪ Requirements analysis	▪ Req'ments development ▪ Requirements mgmt	▪ System requirements analysis and validation
System architecting	System Design ▪ Solution definition	▪ Synthesis	▪ Architectural design ▪ System life cycle mgmt	▪ Select product-component solutions (Tech sol'n) ▪ Develop the design (Tech sol'n)	▪ System product technical req'ments anlys/validation ▪ Design or physical solution representation
System implementation	Product Realization ▪ Implementation ▪ Transition to Use	Not included in scope	▪ Implementation ▪ Integration ▪ Transition	▪ Implement the product design (Tech sol'n) ▪ Product integration	Not included in scope
Technical analysis	Technical Evaluation ▪ Systems analysis	▪ Functional analysis ▪ Requirements trade studies ▪ Functional trade studies ▪ Design trade studies	▪ Requirements analysis	▪ Decision analysis and resolution	▪ Functional analysis, allocations and validation ▪ Assessments of system effectiveness, cost, schedule, and risk ▪ Tradeoff analyses
Technical management/ leadership	Technical Mgmt ▪ Planning ▪ Assessment ▪ Control	▪ Technical mgmt ▪ Track analysis data ▪ Track performance – project plans, tech plans ▪ Track product metrics ▪ Update specifications ▪ Update architectures ▪ Update plans ▪ Maintain database	▪ Planning ▪ Assessment ▪ Control ▪ Decision mgmt ▪ Configuration mgmt ▪ Resource mgmt ▪ Risk mgmt	▪ Project planning ▪ Project monitoring & control ▪ Measurement and analysis ▪ Process and product quality assurance ▪ Configuration mgmt ▪ Integrated project mgmt ▪ Quantitative project mgmt ▪ Risk mgmt	▪ Planning ▪ Monitoring ▪ Decision making, control, and baseline maintenance ▪ Risk mgmt ▪ Baseline change control and maintenance ▪ Interface mgmt ▪ Data mgmt ▪ Technical reviews/audits
Scope management	Acquisition & Supply ▪ Supply ▪ Acquisition	Not included in scope	▪ Acquisition ▪ Supply	▪ Supplier agreement mgmt	▪ Technical mgmt of subcontractors/vendors
Verification & validation	Technical Evaluation ▪ Requirements validation ▪ System verification ▪ End products validation	▪ Requirement verification ▪ Functional verification ▪ Design verification	▪ Verification ▪ Validation	▪ Verification ▪ Validation	▪ Design or physical solution verification and validation
In the standard, but not in agreement with other standards			▪ Operation ▪ Disposal ▪ Enterprise mgmt ▪ Investment mgmt ▪ Quality mgmt	▪ Organ'l process focus ▪ Organ'l process definition ▪ Organ'l training ▪ Organ'l process perf ▪ Causal analysis/resolution ▪ Organ'l innov/deploymnt	▪ Lessons learned and continuous improvement

System Implementation. Some controversy exists over the end products of systems engineering. In some standards, the end products are clearly identified as documentation: architectural depictions, specifications, interface documents, allocated requirements, and design representation. In other standards, the end product is considered to be a functioning system that meets the defined mission or purpose. For the purposes of the SE-ROI project, we take the broader view that systems engineering is responsible for the technical development of the system, usually including delivery and installation of the first or prototype version. With this broader view, there are specific efforts related to the system implementation documented in the standards, such as component implementation, system integration, and transition to use.

Technical Analysis. It is widely accepted that systems engineering is responsible for system-level technical analysis, particularly as related to assessment of system performance against the requirements. Design trade-off analysis is also clearly a systems engineering effort, although there is some conflict as to whether it is part of architecting or is a separate task. An additional type of analysis is sometimes described ("functional analysis") in which the system functions are hierarchically decomposed on functional (as opposed to physical component) boundaries. What characterizes each of these types of analysis is the multi-disciplinary scope of the analysis, focused on system emergent properties.

Technical Management/Leadership. All standards recognize the inherent need for technical management as a part of systems engineering, an effort required by the size of the engineering teams involved in many system design programs. It was noted earlier that the research of [Frank 2000] showed that such management and leadership are widely regarded as attributes of successful systems engineers. The descriptions of these efforts in the standards are significantly different, but can often be interpreted to agree with each other. It is interesting to note that the standards typically expend more effort defining this category of effort than any other, yet this category of effort is sometimes considered by technical practitioners to be outside the scope of SE.

Scope Management. Another aspect of technical management that appears to be somewhat distinct in the standards is the technical definition and management of acquisition and supply issues. Two standards specifically refer to acquisition and supply, while two others explicitly cover subcontract or supplier management. This area of effort applies to the contractual relationships both upward and downward. Upward relationships involve a development contract or internal definition of scope for the entire system development, which contract or scope usually involves the system requirements. Downward relationships involve the contracts or internal scope definition for system components to be developed by others. These relationships are distinct in character from the internal team relationships covered by Technical Management/Leadership.

Verification and Validation. The standards also agree on the inclusion of verification and validation in the scope of systems engineering. Verification is described as the comparison of the system (or the developmental artifacts) with its requirements through the use of examinations, analysis, demonstrations, tests, or other objective evidence. Validation is described as the comparison of the completed system (or artifacts) with the intended mission or purpose of the system.

These eight effort categories, then, appear to describe a useful categorization of systems engineering effort. By collecting data against these categories, the SE-ROI project can seek to understand the correlated impacts of each category of effort on the success of a program.

It should be noted that some standards include other efforts, also shown in Table 5, that are not in general agreement across the standards. While the authors of each standard deemed these efforts suitable for inclusion within the scope of that standard, the lack of general agreement argues against their inclusion in the ontology (the "shared conceptualization"). We also note that two of the five standards (ISO-15288 and CMMI) explicitly intend to cover more than "systems engineering." Three primary areas appear in only one or two of the standards, including:

- System operation and disposal
- Enterprise/organizational management
- Lessons learned and continuous improvement.

Key Resulting Definitions

The categorization and standards review of the previous section lead to some key definitions that are a useful component of the ontology needed.

Systems engineering effort – The standards review shows a working definition of systems engineering effort in terms of the categories discovered: mission/purpose definition, requirements engineering, system architecting, system implementation, technical analysis, technical management/leadership, scope management and verification/validation. These categories will be treated for data collection purposes as the independent variables of the research. Typical working hypotheses will take the form of "There is a quantifiable correlation between the amount/type/quality of requirements engineering effort used during a program and the cost compliance of the program." Each of the underlined phrases might be replaced by other phrases within the ontology. Such a form immediately brings to the forefront the need for further definitions.

Amount – Systems engineering effort can be quantified in terms of the man-hours of effort applied. As shown in [Honour 2004], however, this must also be qualified by a measure of the quality of the effort applied. For comparative purposes, the man-hours of each type of effort will be normalized by the total development man-hours of the project.

Type –This project will explore various types of processes and methods to seek correlations with the program success. The "type" of effort will be characterized by descriptive terms during program interviews. Aggregation of "types" will be performed during statistical analysis. For example, one type of technical analysis might be "the use of software-based Monte Carlo models to predict system performance."

Quality - The quality of systems engineering effort may be largely a matter of the processes and methods used on the program, and the applicability of those processes and methods to the specific program. However, the project will also explore various subjective and objective measures of quality.

Success – The success of a program can be measured in several different ways. Based on the background work, the initially assumed measures include

a. Technical compliance with stakeholder needs, as described in [Browning 2005],
b. Cost compliance of the development program with its budgets,
c. Schedule compliance of the development program with its plans, and
d. Subjective customer/user/developer surveys.

Other success measures will be explored during interviews, including any program-unique success measures.

Optimum – The SE-ROI project seeks to discover the optimum relationships. The optimum will be determined by correlation with program success. Due to the high degree of scatter expected in the data, it is expected that this optimum will be parameterized by various program characteristics.

Future Work in SE-ROI

Based on this preliminary categorization, the SE-ROI project is proceeding to define interview data forms that will allow gathering data in each of the seven categories. Data to be gathered will include:

- Project characterization data such as project size, total development man-hours, project type, development phases, bounding parameters, risk levels.
- Project success data such as cost/schedule compliance and technical quality measures.
- For each of the seven categories, systems engineering data such as
  - Hours expended on systems engineering tasks in the category
  - Quality of those tasks
  - Type of those tasks, the specific nature of the methods and tools used.

Data analysis will use the data gathered to seek correlations that support the hypotheses. This phase uses statistical correlation methods. As each set of data is obtained, the statistical analysis will be extended based on the quality and quantity of the total data set. Initially, there will be insufficient data to reliably support any correlation. With a few data sets, high-level correlations may be attempted. As the number of data sets increases, more correlations may be attempted in accordance with "design of experiments" methods.

Project reports include the generation of interim and final technical reports, in the forms of:

- A public website with summary information. (See http://www.hcode.com/seroi/)
- Interim analysis results, prepared as internal data and distributed to an advisory group
- Benchmark reports, prepared as written reports to each participating organization. The reports will include specific data from the organization's interviewed programs, compared with aggregate data from the project as a whole.
- Final results in the form of a technical dissertation and offered for publication as at least one refereed, journal-level technical paper.

The SE-ROI project has an advisory group that helps to guide the project and obtains early access to the data. The advisory group is open to interested parties. Contact eric.honour@postgrads.unisa.edu.au for information.

Summary

This paper describes the need for a common ontology of systems engineering practices that currently does not exist. The evidence in the widely-used "standards" shows that there is considerable disagreement about the scope of systems engineering, and even more disagreement about the terminology, relationship, and need for various types of effort. Data from the COSYSMO project shows that practitioner organizations have a wide dispersion in their application of the standards.

The SE-ROI project is proceeding to define an ontology as a basis for quantification of systems engineering value. This categorization will be used as a basis for data gathering that can allow correlation of systems engineering activities with program success.

Acknowledgements

The COSYSMO project is sponsored by the Center for Software Engineering (CSE) of the University of Southern California (USC), the Lean Aerospace Initiative (LAI) of the Massachusetts Institute of Technology (MIT), and The Aerospace Corporation. The SE-ROI project is funded and sponsored by Honourcode, Inc. and the Systems Engineering and Evaluation Centre (SEEC) of the University of South Australia (UniSA).

References

ANSI/EIA-632, *Processes for Engineering a System*, American National Standards Institute and Electronics Industries Association, 1999.

Browning, T. and Honour, E., "Measuring the Life Cycle Value of a System," *INCOSE International Symposium*, Rochester, NY, 2005.

CMMI Team, *Capability Maturity Model Integration Version 1.1*, Carnegie Mellon Software Engineering Institute, Pittsburgh, PA, 2002.

Ernstoff, M. and Vincenzini, I., "Guide to Products of System Engineering," *INCOSE International Symposium*, Las Vegas, NV, 1999.

Frank, M. "Cognitive and Personality Characteristics of Successful Systems Engineers," *INCOSE International Symposium*, Minneapolis, MN, 2000

Honour, E.C., "Understanding the Value of Systems Engineering," *Proceedings of the INCOSE International Symposium*, Toulouse, France, 2004.

Honour, E.C., and Valerdi, R., "Toward an Ontology to Measure Systems Engineering Return on Investment," *20th International Forum on COCOMO and Software Cost Modeling*, University of Southern California, Los Angeles, CA, October 2005.

IEEE 1220-1998, *IEEE Standard for Application and Management of the Systems Engineering Process*, Institute of Electrical and Electronics Engineers, 01 May 1998

ISO/IEC-15288:2002, *System Life Cycle Processes*, International Standards Organization, 2002.

Kasser, J., "Systems Engineering: Myth or Reality," *Proceedings of the NCOSE International Symposium*, Boston, MA, 1995.

Pennell, L.W. and Knight, F.L., *Systems Engineering*, Draft MIL-STD-499C, Report number TOR-2005(8583)-3, The Aerospace Corporation, El Segundo, CA, 15 Apr 2005.

Roedler, G., and Rhodes, D., *Systems Engineering Leading Indicators Guide*, Lean Aerospace Initiative (LAI) and INCOSE, Cambridge, MA, 2005.

Sheard, S., "The Shangri-La of ROI," *INCOSE International Symposium*, Minneapolis, MN, 2000.

Shenhar, A. J. and Bonen, Z., "The New Taxonomy of Systems: Toward an Adaptive Systems Engineering Framework", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* vol. 27, no. 2, March 1997, 137 - 145.

Studer, R., V. R. Benjamins, and D. Fensel, "Knowledge Engineering: Principles and Methods." *IEEE Transactions on Data and Knowledge Engineering*, 25 (1-2), pp161-197, 1998.

Valerdi, R., Miller, C., and Thomas, G., "Systems Engineering Cost Estimation by Consensus," *International Conference on Systems Engineering*, Las Vegas, NV, 2004.

Valerdi, R., Wheaton, M., "ANSI/EIA 632 As a Standardized WBS for COSYSMO," *AIAA 1st Infotech@Aerospace Conference*, Arlington, VA, September 2005.

Wynne, M.W., "Policy for Systems Engineering in DoD," Memo by Undersecretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, 20 February 2004.

Biographies

Eric Honour was the 1997 INCOSE President. He has a BSSE from the US Naval Academy and MSEE from the US Naval Postgraduate School, with 37 years of systems experience. He is currently a doctoral candidate at the UniSA SEEC. He was a naval officer for nine years, using electronic systems in P-3 anti-submarine warfare aircraft. He has been a systems engineer, engineering manager, and program manager with Harris, E-Systems, and Link. He has taught engineering at USNA, at community colleges, and in continuing education courses. He was the founding President of the Space Coast Chapter of INCOSE, the founding chair of the INCOSE Technical Board, and a past director of the Systems Engineering Center of Excellence. He was selected in 2000 for Who's Who in Science and Technology and in 2004 as an INCOSE Founder. Mr. Honour provides technical management support and systems engineering training as President of Honourcode, Inc., while continuing research into the quantification of systems engineering.

Ricardo Valerdi is a Research Associate at the Lean Aerospace Initiative at MIT and a Visiting Associate at the Center for Software Engineering at USC. He earned his BS in Electrical Engineering from the University of San Diego, MS and PhD Systems Architecting & Engineering from USC. Formerly he was a Member of the Technical Staff at the Aerospace Corporation in the Economic & Market Analysis Center and a Systems Engineer at Motorola and at General Instrument Corporation.