COSYSMO 3.0
Model for PSMUG
Edited by Jim Alstad
PSMUG Meeting version of 20 February 2016

Top-Level Table of Contents

Overview of this Version

This version was prepared for the PSMUG Meeting of February 25, 2016. It's based on v35 of the working model.

1. (Elided)

2. Non-Driver Model Elements

2.1 Scope

The issue is, what system engineering labor cost elements of a project or program are to be estimated by the model?

2.1.1. Result

The cost elements to be estimated are all the stages in 15288. The stages used in the present model are adjusted for consistency with those in COSYSMO 1.0. (The nominal COSYSMO 1 phases of "Operate, Maintain, or Enhance" and "Replace or Dismantle" are covered in COSYSMO 3.0 under the latter 15288 stages.)

(For reference, section 5.2.2 of [2] says that the in-scope stages for COSYSMO 1.0, adapted from 15288, are Conceptualize, Develop, Operational Test and Evaluation, and Transition to Operation; there is also a paragraph on this topic in section 1.2 of [2] immediately following Figure 1.)

The costs covered by the COSYSMO model are systems engineering labor costs for a particular project for a system-of-interest; costs for subsystems are not included.

This table gives the correspondence between COSYSMO stages and stages from 15288, as specified in chapter 4 of TR 24748-1 [25]:

COSYSMO Stage	Corresponding 15288 Stage
Conceptualize	Concept
Develop	Development, except as covered in the stage below
Operational Test and Evaluation	These Development activities: verifying and validating the system, performing appropriate inspections
Transition to Operation	Utilization activities directed at the transition from development
Production	Production
Utilization	Utilization, except as covered in the "Transition to Operation" stage
Support	Support
Retirement	Retirement

2.2 Top-Level Equation

2.2.1. Result

$$PH = A \cdot (AdjSize)^{E} \cdot \prod_{j=1}^{n} EM_j$$

where *PH* = person-hours and *A* = a calibration factor.

2.3 Size Formula

2.3.1. Result

2.3.1.1. Main Result

$$AdjSize_{C3} = \sum_{SizeDrivers} eReq(Type(SD), Difficulty(SD)) \times PartialDevFactor(AL_{Start}(SD), AL_{End}(SD), RType(SD))$$

$$PartialDevFactor(AL_{Start}, AL_{End}, RType) = \sum_{a=AL_{Start}}^{AL_{End}} ALFraction(a, RType)$$

where:

SD = the particular size driver (requirement, interface, algorithm, or scenario);

eReq = the raw size value looked up in section 4;

AL_x = an Activity Level (summarized in the upper table below and defined in section 2.3.1.2);

RType = the reuse type (DWR (Developed With Reuse) or DFR (Developed For Reuse));

and

ALFraction = a value from the lower table below.

Overview of the Activity Levels

DWR AL	DFR AL	AL #
New	Conceptualized for Reuse	1
Design Modified	(not used)	2
Design Implemented	Designed for Reuse	3
Adapted for Integration	Constructed for Reuse	4
Adopted for Integration	(not used)	5
Managed	Validated for Reuse	6

Values of ALFraction

AL #:	1	2	3	4	5	6
DWR Activity Level:	New	Design Modified	Design Im-plemented	Adapted for Integration	Adopted for Integration	Managed
ALFraction for RType = DWR	14.41%	14.41%	12.38%	19.06%	17.23%	22.52%
DFR Activity Level:	Conceptualized for Reuse	N/A	Designed for Reuse	Constructed for Reuse	N/A	Validated for Reuse
ALFraction for RType = DFR	32.92%		21.99%	26.35%		15.52%

The values shown are inferred from the revised GRF [24].

2.3.1.2. Activity Level Definitions

The material in the following two tables is taken, slightly adapted, from [24].

Definitions of the DWR Activity Levels

DWR AL	Definition	AL #
New	System attribute that is new, which requires developing from scratch; or from previously defined system design or constructed product components but requiring near-complete changes in system architecture as a result of modified or extended system functionalities.	1
Design Modified	System attribute that is designed and developed by leveraging previously defined system concept, functional and logical reference architecture; or from previously designed physical architecture or constructed product components which requires significant design and implementation changes or refactoring but without major changes in system functionalities	2
Design Implemented	System attribute that is implemented from an inherited, completed system design or a previously constructed product component that may require only limited design changes in the physical architecture to an extent that it will not impact or change the basic design but that may require reimplementation of the component.	3
Adapted for Integration	System attribute that is integrated from adaptation or tailoring (by limited modification of interfaces) of previously constructed or deployed product components without changes in the system architecture and design or the physical implementation except for those related to interface changes so that the adapted element can be effectively integrated or form fit into the new system. The effort required is relatively lower than that of the Design Implemented category. This category includes removal of system element from previously developed or deployed system baseline.	4
Adopted for Integration	System attributethat is incorporated or integrated from previously developed or deployed product components without modification, which requires complete integration, assembly, test and checkout activities as well as V&V testing. This is also known as “black-box” reuse or simple integration.	5
Managed	System attributethat is inherited from previously developed and validated product components without modification and the integration of such an element, if required, is through significantly reduced V&V testing effort by means of inspection or provided test services, procedures and equipment. Most of the systems engineering effort incurred is a result of technical management.	6

Definitions of the DFR Activity Levels

DFR AL	Definition	AL #
Conceptualized for Reuse	This Activity Level encapsulates a set of front-end systems engineering activities from which reusable resource produced is a logical or functional architecture that must be further developed through a series of detailed design, implementation, verification and validation testing activities to realize the final deployable product.	1
Designed for Reuse	This Activity Level encapsulates a set of front-end of system design activities from which reusable resource produced is a complete system design or physical architecture that must be further developed through a series of implementation, integration, verification and validation testing activities to realize the final deployable product.	3
Constructed for Reuse	This Activity Level encapsulates a set of system development activities from which reusable resource produced is a physical product or component that has been implemented and independently verified through verification testing but has not been deployed or used in an end system. This requires all levels of system development activities short of final system-level integration, transition, verification and validation testing.	4
Validated for Reuse	This Activity Level encapsulates the entire set of system development activities from which reusable resource produced is a physical product or component that has been developed, deployed, and operational validated through its use in an end system.	6

2.4 Exponent Formula

2.4.1. Result

$$E = E_{COSYSMO1} + SF_{ROR} + SF_{PC} + SF_{RV}$$

where:

$E_{COSYSMO1}$ = 1.06; and

SFx = the values of the scale factors from section 5.

2.5 Grouping of Cost Drivers

2.5.1. Result

Understanding Factors
CONOPS and Requirements Understanding
Architecture Understanding
Stakeholder Team Cohesion

Complexity Factors
Level of Service Requirements
Technology Risk
# of Recursive Levels in the Design
DFR

Operations Factors
and Diversity of Installations/Platforms
Migration Complexity
Interoperability

People Factors
Personnel/Team Capability
Process Capability
Personnel Experience/Continuity

Environment Factors
Multisite Coordination
Tool Support

2.6 (Elided)

2.7 Allowed Textual Levels for Cost Drivers and Scale Factors

2.7.1. Result

There is a 6-level textual rating scale available, from Very Low through Extra High. (However, for a particular parameter not all levels need to be used.)

When the definition of a driver is the same in the current Model as in COSYSMO 1.0 (or in other starting model), the textual rating levels are also the same.

When the definitions differ, it is straightforward to map a COSYSMO 1.0 rating level to a current Model rating level.

2.8 Allowed Numeric Levels for Cost Drivers and Scale Factors

2.8.1. Result

(This assumes a straightforward mapping of textual levels to integers.)

The allowed numeric levels are either:

- The integer corresponding directly to a textual level; or
- A number that is 25%, 50%, or 75% of the way from one allowed integer to another.

2.9 (Elided)

2.10 Applicability to a System-of-Systems

2.10.1. Result

Applies to a system-of-systems.

2.11 (Elided)

2.12 Account for Correlations among Driver Values

2.12.1. Result

Correlations among drivers should not be considered when defining the model. If correlation

among drivers appears during model calibration, that can be addressed as a calibration problem.

2.13 How To Handle Too Much Variance

2.13.1. Result

If the model makes good predictions (per the accepted statistical criteria), there is no such thing as "too much variance".

2.14 (Elided)

2.15 Multi-Subproject Model

2.15.1. Result

The model from Jo Ann's SoS estimating paper [10] is used:

$$PM_{C3M} = A_{C3} \cdot (TotalSize_{C3})^{E_{C3}} \cdot \sum_{s \in Subprojects} \left(\frac{Subproject_s Size_{C3}}{TotalSize_{C3}} \cdot \prod_{j=1}^{15} EM_{C3:s,j} \right)$$

2.16 Project versus Product Life Cycle

2.16.1. Result

COSYSMO estimates project life cycle costs (in part, per section 2.1).

2.17 "Effort Multiplier" or "Cost Driver"?

This is a terminology issue: Should elements in section 3 (formerly titled "Cost Drivers") be called "effort multipliers" or "cost drivers"?

2.17.1. Result

This use of terms was agreed to:

Effort Multiplier: a number (as suggested by Dan): a rating value.

Cost Driver: term for the "variable", as in COCOMO.

Driver: An inclusive term, covering cost drivers, size drivers, scale factors, and reuse factors.

Model Element (or just "Element"): either a driver, or some other defining aspect of the model, such as those covered in section 2.

3. Cost Drivers

3.1 CONOPS and Requirements Understanding

In v13, Requirements Understanding (**Error! Reference source not found.**) has been merged into the present section. The old section has been maintained so as to record its past discussions.

3.1.1. Definition

Text Definition: The extent to which the Stakeholders and Team understand the system's concept of operations. In addition, this cost driver rates the level of understanding of the system requirements by all stakeholders including systems, software, hardware, customers, team members, users, etc. Primary sources of added systems engineering effort are unprecedented systems, unfamiliar domains, or systems whose requirements are emergent with use.

Rating Scale:

Viewpoints:	Very low	Low	Nominal	High	Very High
Degree of Understand-ing of CONOPS	The Stakeholders and Team have a poor understanding of the CONOPS.	The Stakeholders and Team have a mediocre understanding of the CONOPS.	The Stakeholders and Team have an average understanding of the CONOPS.	The Stakeholders and Team have a good understanding of the CONOPS.	The Stakeholders and Team have a strong understanding of the CONOPS.
Unresolved Issues in the CONOPS	Multiple critical issues are unresolved in Stakeholders' understanding of the CONOPS	One critical issue is unresolved in Stakeholders' understanding of the CONOPS	No critical issues are unresolved in Stakeholders' understanding of the CONOPS, but several significant issues are unresolved	No critical issues are unresolved in Stakeholders' understanding of the CONOPS, but a few significant issues are unresolved	No critical or significant issues are unresolved in Stakeholders' understanding of the CONOPS
Requirements Under-standing	Poor: emergent requirements or unprecedented system	Minimal: many undefined areas	Reasonable: some undefined areas	Strong: few undefined areas	Full understanding of requirements, familiar system
User Training	The user community has received little or no training on the new/modified system,	The user community has received a less than average amount of training on the new/modified	The user community has received an average amount of training on the new/modified	The user community has received a greater than average amount of training on the	The user community has received a superior amount of training on the new/modified

	including on any new technology in use.	system, including on any new technology in use.	system, including on any new technology in use.	new/modified system, including on any new technology in use.	system, including on any new technology in use.

Guidance: Whenever the Requirements Volatility scale factor is rated above Very Low, the Requirements Understanding viewpoint should be taken to be "Nominal"; the motivation is to avoid "double-counting" requirements volatility.

EMR: Mean = 3.088, standard deviation = 0.290.

3.2 (Elided)

3.3 Architecture Understanding

3.3.1. Definition

Text Definition: This cost driver rates the degree of understanding of determining and managing the system architecture in terms of platforms, standards, new and NDI (COTS/GOTS) components, connectors (protocols), and constraints. This includes tasks like systems analysis, tradeoff analysis, modeling, simulation, case studies, etc.

Rating Scale:

Very low	Low	Nominal	High	Very High
Poor understanding of architecture and NDI, unprecedented system	Minimal understanding of architecture and NDI, many unfamiliar areas	Reasonable understanding of architecture and NDI, some unfamiliar areas	Strong understanding of architecture and NDI, few unfamiliar areas	Full understanding of architecture, familiar system and NDI

EMR: Mean = 2.448, standard deviation = 0.022.

3.4 Stakeholder Team Cohesion

3.4.1. Definition

Text Definition: Represents a multi-attribute parameter, which includes leadership, shared vision, diversity of stakeholders, approval cycles, group dynamics, Integrated Product Team 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.

Rating Scale:

	Very Low	**Low**	**Nominal**	**High**	**Very High**
Culture	Stakeholders with diverse expertise, task nature, language, culture, infrastructure Highly heterogeneous stakeholder communities	Heterogeneous stakeholder community Some similarities in language and culture	Shared project culture	Strong team cohesion and project culture Multiple similarities in language and expertise	Virtually homogeneous stakeholder communities Institutionalized project culture
Compatibility	Highly conflicting organizational objectives	Converging organizational objectives	Compatible organizational objectives	Clear roles & responsibilities	Strong mutual advantage to collaboration
Familiarity and Trust	Complete lack of familiarity	Willing to collaborate, little familiarity	Some familiarity and trust	Extensive successful collaboration	Very high level of familiarity and trust

EMR: Mean = 2.419, standard deviation = 0.233.

3.5 Level of Service Requirements

3.5.1. Definition

Text Definition: This cost driver rates the difficulty and criticality of satisfying the ensemble of level of service requirements, such as security, safety, response time, maintainability, Key Performance Parameters (KPPs), system qualities (formerly known as the "ilities"), etc.

Rating Scale:

	Very low	**Low**	**Nominal**	**High**	**Very High**
Difficulty	Simple; single dominant KPP	Low, some coupling among KPPs	Moderately complex, coupled KPPs	Difficult, coupled KPPs	Very complex, tightly coupled KPPs
Criticality	Slight inconvenience	Easily recoverable losses	Some loss	High financial loss	Risk to human life

EMR: Mean = 2.625, standard deviation = 0.187.

3.6 Technology Risk

3.6.1. Definition

Text Definition: The maturity, readiness, and obsolescence of the technology being implemented. Immature or obsolescent technology will require more Systems Engineering effort.

Rating Scale:

	Very Low	Low	Nominal	High	Very High
Lack of Maturity/ Readiness	Technology proven and widely used throughout industry. Mission proven.	Proven through actual use and ready for widespread adoption. Concept	Proven on pilot projects and ready to roll- out for production jobs. Concept has been	Ready for pilot use. Proof of concept validated. (TRL 5 & 6)	Still in the laboratory. Concept defined. (TRL 3 & 4)

	(TRL 9)	qualified. (TRL 8)	demonstrated. (TRL 7)		
Obsolescence			- Technology is the state-of-the-practice - Emerging technology could compete in future	- Technology is stale - New and better technology is ready for pilot use	- Technology is outdated and use should be avoided in new systems - Spare parts supply is scarce

EMR: Mean = 2.654, standard deviation = 0.251.

3.7 # of Recursive Levels in the Design

3.7.1. Definition

Text Definition: The number of levels of design related to the system-of-interest (as defined by ISO/IEC 15288.

Rating Scale:

	Very Low	Low	Nominal	High	Very High
Number of levels	1	2	3-5	6-7	>7

EMR: Mean = 2.119, standard deviation = 0.381.

3.8 # and Diversity of Installations/Platforms

3.8.1. Definition

Text Definition: The number of different platforms that the system will be hosted and installed on. The complexity in the operating environment (space, sea, land, fixed, mobile, portable, information assurance/security, constraints on size weight, and power). For example, in a wireless network it could be the number of unique installation sites and the number of and types of fixed clients, mobile clients, and servers. Number of platforms being implemented should be added to the number being phased out (dual count).

Rating Scale:

	Nominal	High	Very High	Extra High
Sites/ installations	Single installation site or configuration	2-3 sites or diverse installation configurations	4-5 sites or diverse installation configurations	>6 sites or diverse installation configurations
Operating environment	Existing facility meets all known environmental operating requirements	Moderate environmental constraints; controlled environment (i.e., A/C, electrical)	Ruggedized mobile land-based requirements; some information security requirements. Coordination between 1 or 2 regulatory or cross functional agencies required.	Harsh environment (space, sea airborne) sensitive information security requirements. Coordination between 3 or more regulatory or cross functional agencies required.
Platforms	<3 types of platforms being installed and/or being phased out/replaced	4-7 types of platforms being installed and/or being phased out/replaced	8-10 types of platforms being installed and/or being phased out/replaced	>10 types of platforms being installed and/or being phased out/replaced
	Homogeneous platforms	Compatible platforms	Heterogeneous, but compatible platforms	Heterogeneous, incompatible platforms
	Networked using a single protocol	Networked using a single protocol and multiple operating systems	Networked using a mix of protocols; single operating system	Networked using a mix of protocols; multiple operating systems

EMR: Mean = 1.842, standard deviation = 0.173.

3.9 Migration Complexity

3.9.1. Definition

Text Definition: This cost driver rates the extent to which the legacy system affects the migration complexity, if any. Legacy system components, databases, workflows, environments, etc., may affect the new system implementation due to new technology introductions, planned upgrades, increased performance, business process reengineering, etc.

Rating Scale:

	Nominal	High	Very High	Extra High
Legacy contractor	Self; legacy system is well documented. Original team largely available	Self; original development team not available; most documentation available	Different contractor; limited documentation	Original contractor out of business; no documentation available
Effect of legacy system on new system	Everything is new; legacy system is completely replaced or non-existent	Migration is restricted to integration only	Migration is related to integration and development	Migration is related to integration, development, architecture and design

EMR: Mean = 2.068, standard deviation = 0.418.

3.10 Interoperability

3.10.1. Definition

Text Definition: How extensive are the interoperability requirements? Interoperability is defined as "The ability of a system to work with another system or group of systems". External interoperability (interoperability with other systems) is always considered. When the system of

interest is a system-of-systems, internal interoperability (interoperability between constituent systems) also applies.

Rating Scales:

There are two different rating scales; the appropriate one should be selected, depending on whether the project is for an existing system or for a new system.

Existing System Rating Scale:

The existing system External Interoperability scale is based on LISI levels [15]; these ratings are to be interpreted this way: "Before being integrated into a system-of-systems, what was the system's status with regard to interoperability?".

Viewpoint	Very Low	Low	Nominal	High	Very High
External Interoperability	Isolated.	Connected.	No interoperability requirements; or functional standards employed.	Domain standards employed.	Enterprise standards employed.
Internal Interoperability	There are a very large number of significant inconsistency/ incompatibility issues in standards, databases, and interfaces among constituent systems.	There are a large number of significant inconsistency/ incompatibility issues in standards, databases, and interfaces among constituent systems.	This is not a system-of-systems; or there are a moderate number of significant inconsistency/ incompatibility issues in standards, databases, and interfaces among constituent systems.	There are a few significant inconsistency/ incompatibility issues in standards, databases, and interfaces among constituent systems.	There are no significant inconsistency/ incompatibility issues in standards, databases, and interfaces among constituent systems.

New System Rating Scale:

The new system External Interoperability scale is based on LCIM conceptual levels [16].

Viewpoint	Very Low	Low	Nominal	High	Very High
External Interoperability	System-specific data.	Documented data.	No interoperability requirements; or aligned static data.	Aligned dynamic data.	Harmonized data.
Internal Interoperability	Existing constituent systems do not interoperate, and they are large and complex.	Existing constituent systems do not interoperate, but they are simple and/or small.	This is not a system-of-systems; or all constituent systems are new; or all existing constituent systems presently interoperate.		

EMR: Mean = 2.045, standard deviation = 0.256.

3.11 Personnel/Team Capability

3.11.1. Definition

Text Definition: Composite systems engineering capability of a team of Systems Engineers (compared to the national pool of SEs) on the attributes of analyzing complex problems and synthesizing solutions, being efficient and thorough, and having the ability to communicate and cooperate.

Rating Scale:

Very Low	Low	Nominal	High	Very High
15th percentile	35th percentile	55th percentile	75th percentile	90th percentile

EMR: Mean = 2.530, standard deviation = 0.543.

3.12 Process Capability

3.12.1. Definition

Text Definition: The consistency and effectiveness of the project team at performing SE processes. This may be based on assessment ratings from a published process model (e.g., CMMI, EIA-731, SE-CMM, ISO/IEC15504). It can alternatively be based on project team behavioral characteristics, if no assessment has been made.

Rating Scale:

	Very Low	Low	Nominal	High	Very High	Extra High
Assessment Rating	Level 0 (if contin-uous model)	Level 1	Level 2	Level 3	Level 4	Level 5
Project Team Behavioral Characteristics	Ad Hoc approac h to process perform ance	Performed SE process, activities driven only by immediate contractual or customer requirements, SE focus limited	Managed SE process, activities driven by customer and stakeholder needs in a suitable manner, SE focus is requirements through design, project- centric approach – not driven by organizational processes	Defined SE process, activities driven by benefit to project, SE focus is through operation, process approach driven by organizational processes tailored for the project	Quantitatively Managed SE process, activities driven by SE benefit, SE focus on all phases of the life cycle	Optimizing SE process, continuous improvement, activities driven by system engineering and organizational benefit, SE focus is product life cycle & strategic applications
SEMP Sophistication	Manage ment judgmen t is used	SEMP is used in an ad-hoc manner only on portions of the project that require it	Project uses a SEMP with some customization	Highly customized SEMP exists and is used throughout the organization	The SEMP is thorough and consistently used; organizational rewards are in place for those that improve it	Organization develop best practices for SEMP; all aspects of the project are included in the SEMP; organizational rewards exist for those that improve it

EMR: Mean = 2.530, standard deviation = 0.543.

3.13 Personnel Experience/Continuity

3.13.1. Definition

Text Definition: The applicability and consistency of the staff at the initial stage of the project with respect to the domain, customer, user, technology, tools, etc.

Rating Scale:

	Very low	Low	Nominal	High	Very High
Experience	Up to 1 year experience	3 years of continuous experience	5 years of continuous experience	10 years of continuous experience	20 years of continuous experience
Annual Turnover	48%	24%	12%	6%	3%

EMR: Mean = 2.262, standard deviation = 0.301.

3.14 Multisite Coordination

3.14.1. Definition

Text Definition: Location of stakeholders, team members, resources, corporate collaboration barriers.

Rating Scale:

	Very Low	Low	Nominal	High	Very High	Extra High
Collocation	International, severe time zone impact	Multi-city and multi-national, considerable time zone impact	Multi-city or multi-company, some time zone effects	Same city or metro area	Same building or complex, some co-located stakeholders or onsite representation	Fully co-located stakeholders
Commun-ications	Some phone, mail	Individual phone, FAX	Narrowband e-mail	Wideband electronic communication	Wideband electronic communication, occasional video conference	Interactive multimedia
Corporate Collaboration Barriers	Severe export and security restrictions	Mild export and security restrictions	Some contractual & Intellectual property constraints	Some collaborative tools & processes in place to facilitate or overcome, mitigate barriers	Widely used and accepted collaborative tools & processes in place to facilitate or overcome, mitigate barriers	Virtual team environment fully supported by interactive, collaborative tools environment

EMR: Mean = 2.061, standard deviation = 0.329.

3.15 Tool Support

3.15.1. Definition

Text Definition: Coverage, integration, and maturity of the tools in the Systems Engineering environment.

Rating Scale:

Very low	Low	Nominal	High	Very High
No SE tools, or simple SE tools with little integration.	Basic SE tools moderately integrated throughout the systems engineering process	Strong, mature SE tools, moderately integrated with other disciplines. Cover many parts of the life cycle.	Strong, mature domain model-based life cycle tools. Cover all important parts of the life cycle. Strong model and consistency checking, integration with management tools.	Very strong, mature, domain model-based, knowledge-based life cycle tools. Cover the complete life cycle. Thorough integration across life cycle and management tools. Advanced knowledge-based diagnosis of leading risk indicators

EMR: Mean = 1.838, standard deviation = 0.255.

3.16 (Elided)

3.17 DFR

This cost driver is part of the Harmonized Reuse Model (section 6.3); see that section for background.

3.17.1. Definition

Text Definition: Is the project (or subproject) developing artifacts to be reused on later project(s)? ("Development for Reuse", or "DFR".) If so, what is the extent of the planned reuse?

Rating Scale:

Low	Nominal	High	Very High	Extra High
No reuse at all.	Artifacts will be reused only on the current project.	Artifacts will be reused across the program.	Artifacts will be reused across a product line.	Artifacts will be reused across multiple product lines.

EMR: Mean = 1.798, standard deviation = 0.285.

4. Size Drivers

The model assumes that all size drivers are counted.

4.1 System Requirements

4.1.1. Definition

Text Definition: This driver represents the number of requirements for the system-of-interest at the system level or the level of "sell-off" to the customer, which may include derived requirements at the same Level. The quantity of requirements includes those related to the effort involved in engineering the system interfaces, system specific algorithms, and operational scenarios. 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. Each requirement must have systems engineering effort associated with it such as verification and validation, functional decomposition, functional allocation, etc. System requirements can typically be quantified by counting the number of applicable "shalls" in the system or marketing specification. Note on "shall": that word is used by the US Department of Defense to flag requirements. In other contexts other words may be used for this purpose, such as "will", "must", "should", "may", or "provides"; use a consistent word or combination of words appropriate to your context.

Rating Scale:

Easy	Nominal	Difficult
Simple to implement	**Moderately difficult to implement**	Complex to implement or engineer
Traceable to source	Can be traced to source with some effort	Hard to trace to source
Little requirements overlap	Some overlap	High degree of requirements overlap

Impacts (eReqs/Standard deviation):

Easy	Nominal	Difficult
0.52/0.08	1.00	4.52/0.62

4.2 System Interfaces

4.2.1. Definition

Text Definition: This driver represents the number of shared physical and logical boundaries between system components or functions (internal interfaces) and those external to the system (external interfaces). These interfaces typically can be quantified by counting the number of external and internal system interfaces among ISO/IEC 15288-defined system elements.

Rating Scale:

Easy	**Nominal**	**Difficult**
Simple & straightforward	Moderate complexity	Complex protocol(s)
Uncoupled	Loosely coupled	Highly coupled
Strong consensus	Moderate consensus	Low consensus
Well behaved	Predictable behavior	Poorly behaved
Domain or enterprise standards employed	*Functional standards employed*	*Isolated or connected systems with few or no standards*

The final, italicized row is part of accounting for interoperability; see the 7/03 Alternative below for more information.

Impacts without interoperability row (eReqs/Standard deviation):

Easy	Nominal	Difficult
1.06/0.05	2.67/0.28	6.07/0.46

Impacts with interoperability row (eReqs/Standard deviation):

Easy	Nominal	Difficult
1.92/0.53	3.93/0.81	9.01/1.64

4.3 Algorithms

4.3.1. Definition

Text Definition: This driver represents the number of mathematical algorithms to be derived in order to achieve the system functional and performance requirements. The number can be quantified by counting the number of unique algorithms needed to realize key system requirements specified in the system specification or architecture description document. As an example, this could include a complex aircraft tracking algorithm like a Kalman Filter being derived using existing experience as the basis for the all aspect search function. Another example could be a discrimination algorithm being derived to identify friend or foe function in space-based applications.

Rating Scale:

Easy	Nominal	Difficult
- Algebraic	- Straight forward calculus	- Complex constrained optimization; pattern recognition
- Straightforward structure	- Nested structure with decision logic	- Recursive in structure with distributed control

- Simple data	- Relational data	- Noisy, ill-conditioned data
- Timing not an issue	- Timing a constraint	- Dynamic, with timing and uncertainty issues
- Adaptation of library-based solution	- Some modeling involved	- Simulation and modeling involved

Impacts (eReqs/Standard deviation):

Easy	Nominal	Difficult
1.96/0.21	3.87/0.37	10.00/1.19

4.4 Operational Scenarios

4.4.1. Definition

Text Definition: This driver represents the number of operational scenarios that a system must satisfy in order to accomplish its intended mission or mission objectives. An operational scenario must be end-to-end and triggered by an operational event. Such scenarios include both the nominal stimulus-response thread plus all of the off-nominal threads resulting from bad or missing data, unavailable processes, or other exceptional conditions. The number of scenarios can typically be quantified by counting the number of system-level use cases developed as part of the operational architecture or by counting operational modes captured in the user manual.

Rating Scale:

Easy	Nominal	Difficult
- Well defined	- Loosely defined	- Ill defined
- Loosely coupled	- Moderately coupled	- Tightly coupled or many dependencies/conflicting requirements

- Timelines not an issue	- Timelines a constraint	- Tight timelines through scenario network
- Few, simple off-nominal threads	- Moderate number or complexity of off-nominal threads	- Many or very complex off-nominal threads

Impacts (eReqs/Standard deviation):

Easy	Nominal	Difficult
6.41/1.17	13.83/2.09	26.10/4.32

4.5 (Elided)

5. Scale Factors

5.1 (Elided)

5.2 Risk/Opportunity Resolution

5.2.1. Definition

Text Definition: This driver captures the project's use of a comprehensive, effective risk management process.

Rating Scale:

Viewpoint	Very Low	Low	Nominal	High	Very High	Extra High
A life cycle-long, funded process for identifying, tracking, and resolving risks is carried out.	No such process, or the process is very weak.	The process is weak.	The process is moderate.	The process is fairly strong.	The process is strong.	The process is very strong.
A culture of risk identification, tracking, and resolution is part of the organization.	Very weak culture.	Weak culture.	Moderate culture, including experience in risk management.	Fairly strong culture, including fairly successful experience in risk management.	Strong culture, including mostly successful experience in risk management.	Very strong culture, including very successful experience in risk management.

Maximum value: Mean = 0.06631, standard deviation = 0.01266.

Resulting rating table:

Rating	Very Low	Low	Nominal	High	Very High	Extra High

Exponent value for Risk and Opportunity Resolution	0.06631	0.05305	0.03979	0.02652	0.01326	0.00000

5.3 Process Capability

5.3.1. Definition

The definition is given in section 3.12.1.

Maximum value: Mean = 0.06199, standard deviation = 0.01311.

Resulting rating table:

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Exponent value for Process Capability	0.06199	0.04959	0.03719	0.02480	0.01240	0.00000

5.4 Requirements Volatility

5.4.1. Definition

Text Definition: Requirements volatility is defined as unplanned changes in requirements over a

given set of process stages during the system's life cycle. These changes may include additions, modifications or deletions.

Rating Scale:

Characteristic	Very Low < 1.5	Low >1.5-2.5	Moderate >2.5-3.5	High >3.5-4.5	Very High > 4.5	Weight
System requirements baselined and agreed to by key stakeholders	Fully 1	Mostly 2	Generally 3	Somewhat 4	No Agreement 5	26%
Level of uncertainty in key customer requirements, mission objectives, and stakeholder needs	Very Low 1	Low 2	Moderate 3	High 4	Very High 5	22%
Number of co-dependent systems with influence on system requirements	Very Low 1	Low 2	Moderate 3	High 4	Very High 5	16%
Strength of your organization's requirements development process and level of change control rigor	Very High 1	High 2	Moderate 3	Low 4	Very Low 5	8%
Precedentedness of the system, use of mature technology	Very High 1	High 2	Moderate 3	Low 4	Very Low 5	9%
Stability of stakeholders' organizations (developer, customer)	Very High 1	High 2	Moderate 3	Low 4	Very Low 5	14%
Experience level of the systems engineering team in requirements analysis and development	Very High 1	High 2	Moderate 3	Low 4	Very Low 5	6%

Maximum value: Mean = 0.03518, standard deviation = 0.02170.

Resulting rating table:

Rating	Very Low	Low	Nominal	High	Very High
Exponent value for Requirements Volatility	0.00000	0.00880	0.01759	0.02638	0.03518

6. Reuse Factors

6.1 (Elided)

6.2 (Elided)

6.3 Harmonized Reuse Model

Jim has proposed a model that covers both DWR and DFR (section 3.17). A presentation that covers and discusses this submodel is Jim's "Harmonized_vs_Generalized.v3.pptx", 11/10

6.3.1. Definition

The key parts of the definition (copied from the presentation) are covered in sections 2.3 and 3.17, primarily, and 2.2.

7. Bibliography

1. "A Generalized Systems Engineering Reuse Framework and its Cost Estimating Relationship", Gan Wang, Garry J Roedler, Mauricio Pena, and Ricardo Valerdi, submitted for publication.

2. "The Constructive Systems Engineering Cost Model (COSYSMO)", Ricardo Valerdi (PhD Dissertation), 2005.

3. "Estimating Systems Engineering Reuse with the Constructive Systems Engineering Cost Model (COSYSMO 2.0)", Jared Fortune (PhD Dissertation), 2009.

4. "Quantifying the Impact of Requirements Volatility on Systems Engineering Effort", Mauricio Pena (PhD Dissertation), 2012.

5. "Life Cycle Cost Modeling and Risk Assessment for 21st Century Enterprises", Barry Boehm, Jo Ann Lane, Supannika Koolmanojwong, Richard Turner (presentation), April 29, 2014.

6. "System Interoperability Influence on System of Systems Engineering Effort", Jo Ann Lane, Ricardo Valerdi, CSER 2011.

7. "COSYSMO Extension as a Proxy Systems Cost Estimation" (presentation), Reggie Cole, Garry Roedler, October 23, 2013.

8. "COSATMO: Developing Next-Generation Full-Coverage Cost Models" (presentation), Jim Alstad, USC CSSE Annual Research Review, April 29, 2014.

9. "Quantifying the Impact of Requirements Volatility on Systems Engineering Effort" (presentation), Mauricio Peña, Ricardo Valerdi, October 18, 2012 (COCOMO Forum).

10. "Cost Model Extensions to Support Systems Engineering Cost Estimation for Complex Systems and Systems of Systems", Jo Ann Lane, CSER 2009.

11. "Proposed Modification to COSYSMO Estimating Relationship", Gan Wang, Ricardo Valerdi, Barry Boehm, Alex Shernoff, INCOSE 2008.

12. "Towards COSYSMO 3.0", revised PowerPoint presentation by Gan Wang (filename is "COSYSMO 3.0 Definition Outline.Rev.pptx"), May 2015.

13. "COSATMO: Presenting the Harmonized COSYSMO 3.0 Model" (presentation), Jim Alstad, October 22, 2014 (COCOMO Forum).

14. "Towards COSYSMO 3.0", 3rd revision of PowerPoint presentation by Gan Wang (revises [12]) (filename is "COSYSMO 3.0 Definition Outline.Rev3.pptx"), June 2015.

15. "C4ISR Architecture Working Group Final Report - Levels of Information System Interoperability (LISI)", Department of Defense, Washington DC: OSD(ASD(C3I)) C4ISR AWG, 1998.

16. "The Levels of Conceptual Interoperability Model", Tolk, A., and Muguira, J., Proceedings of the 2003 Fall Simulation Interoperability Workshop. Orlando FL, September 2003.

17. "Towards a Holistic, Total Engineering Cost Model", Wang, G., Shernoff, A., Turnidge, J., and Valerdi, R., INCOSE Singapore, July 2009.

18. "COSYSMO Reuse Extension", Wang, G., Valerdi, R., Ankrum, A., Millar, C., and Roedler, G., INCOSE Utrecht, June 2008.

19. "COSATMO: Presenting the Harmonized COSYSMO 3.0 Model" (presentation), Jim Alstad, October 22, 2014 (COCOMO Forum)

20. "A Survey on Interoperability Measurement." Ford, T., Colombi, J., Graham, S., and Jacques, D., Twelfth International Command and Control Research and Technology Symposium (12th ICCRTS), Newport, RI, June 19-21 2007.

21. "A Framework For Intelligent Assessment and Resolution of Commercial-Off-The-Shelf Product Incompatibilities", Bhuta, Jesal (PhD Dissertation), 2007.

22. "Software Engineering Economics" Boehm, Barry W., Prentice-Hall, Upper Saddle River NJ, 1981.

23. "The Incremental Commitment Spiral Model", Boehm, B., Lane, J., Koolmanojwong, S., Turner, R., Addison-Wesley, Upper Saddle River, NJ, 2014.

24. "The Generalized Reuse Framework—Strategies and the Decision Process for Planned Reuse", Wang, G., INCOSE SEDC16.

25. "Systems and Software Engineering—Life Cycle Management—Part 1: Guide for Life Cycle Management", ISO/IEC TR 24748-1:2010(E).