Development of COSYSMO 3.0: An Extended, Unified Cost Estimating Model for Systems Engineering

Jim Alstad

University of Southern California
Development of COSYSMO 3.0: An Extended, Unified Cost Estimating Model for Systems Engineering

Jim Alstad  USC Center for Systems and Software Engineering

CSER 2019
National Press Club
Washington DC USA
April 3 2019

Jim Alstad  310/344-0894  jalstad@usc.edu
Agenda

Agenda:

• Preliminaries: motivation, history, research hypothesis, & methodology
• The priors; the COSYSMO 3.0 formula
• Closing: Developing the Final Model; future research
• Appendices: Bibliography; backup slides
COSYSMO 3.0 Motivation

• **Context:**
  – Current and future trends create challenges for full-system cost estimation
    • Emergent requirements, rapid change, net-centric systems of systems, COTS, clouds, apps, widgets, high assurance with agility, multi-mission systems
  – Current development practices can minimize cost of one phase, such as development, while raising full-system cost

• **COSYSMO 3.0 is being developed to mitigate this situation by supporting accurate estimates of systems engineering costs, with benefits including:**
  – Allowing thoughtful system-level systems engineering during development, which can result in, for example, choosing new technologies that reduce total system cost
  – Allowing thoughtful engineering of systems to support life-cycle flexibility
History of COSYSMO Models

COSYSMO 1.0
Valerdi, 2005
- Identifies form of model
- Identifies basic cost drivers
- Identifies Size measure

With Reuse
Wang et al, 2008
- Adds weights to Size elements, reducing net Size in the presence of reuse

Req’ts Volatile
Pena, 2012
- Adds scale factor based on requirements volatility

For Reuse
Wang et al, 2014
- Adds weights to Size elements, reducing net Size when artifacts are only partially completed

Sys of Sys
Lane et al, 2011
- Allocates SE effort to SoS and constituent systems. Adds effort multiplier when in the presence of system-of-systems.

COSYSMO 3.0
Alstad, 2018
- Integrates features of previous models
Research Hypothesis

It is possible to develop a systems engineering cost estimating model ("COSYSMO 3.0") with these properties:

- Is applicable to a wide range of systems engineering projects;
- Includes all the major features of COSYSMO 1.0 and its extension models, except for interoperability;
- Provides continuity to users of previous COSYSMO-family models;
- When calibrated to data from a particular organization, estimates actual systems engineering costs with a PRED(.30) accuracy of 50%.
Determine Model Needs

Step 1

Analyze existing literature

Perform Behavioral analyses

Step 2

Define relative significance, data, ratings

Perform expert judgment Delphi assessment, formulate a priori model

Step 3

Gather project data

Determine Bayesian A-Posteriori model

Step 4

Gather more data; refine model

Step 5

Step 6

Step 7

Step 8

This step yielded the Expert-Based Model

Figure 4.1 from [22]
Agenda:

• Preliminaries: motivation, history, research hypothesis, & methodology
• The priors; the COSYSMO 3.0 formula
• Closing: Developing the Final Model; future research
• Appendices: Bibliography; backup slides
Developing the Prior Model

I led COSYSMO 3.0 Wideband Delphi workshops at 4 conferences over the period August 2015-March 2016

• Purpose: To develop consensus expert opinion on the numerical value of COSYSMO 3.0 parameters
• Delphi = anonymous voting; Wideband = group discussion
• Protocol for a parameter:
  – I state a starting value for a parameter
  – Each expert fills in a paper ballot with a recommended value
  – I collect ballots & announce average value
  – I lead a discussion; if there is not consensus, another round of voting is held
• Resulted in 26 expert ballots
  – Experts represented 21 organizations
  – With 19.6 average years of SysEng experience
• Result was “Expert-Based COSYSMO 3.0”
Elements of the COSYSMO 3.0 model:

- Calibration parameter A
- Adjusted Size model
  - eReq submodel, where 4 products contribute to size
  - Reuse submodel
- Exponent (E) model
  - Accounts for diseconomy of scale
  - Constant and 3 scale factors
- Effort multipliers EM
  - 13 cost drivers

\[ PH = A \times (Adj\text{Size})^E \times \prod_{j=1}^{13} EM_j \]
COSYSMO 3.0 Size Model

\[
AdjSize_{C3} = \sum_{SizeDrivers} eReq(\text{Type}(SD), \text{Difficulty}(SD)) \times \text{PartialDevFactor}(A_{L_{\text{Start}}}(SD), A_{L_{\text{End}}}(SD), R_{\text{Type}}(SD))
\]

- **SizeDriver** is one of the system engineering products that determines size in the COSYSMO family (per [2]). Any product of these types is included:
  - System requirement
  - System interface
  - System algorithm
  - Operational scenario

- There are two submodels:
  - Equivalent nominal requirements ("eReq")
    - Raw size
  - Partial development
    - Adjusts size for reuse
Size Model – eReq Submodel

- The eReq submodel is unchanged from [2].
- The submodel computes the size of a SizeDriver, in units of eReq (“equivalent nominal requirements”)
- Each SizeDriver is evaluated as being easy, nominal, or difficult.
- The following table contains conversion factors for the conversion of a SizeDriver to a number of eReq:

<table>
<thead>
<tr>
<th>Size Driver Type</th>
<th>Easy</th>
<th>Nominal</th>
<th>Difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Requirement</td>
<td>0.5</td>
<td>1.0</td>
<td>4.5</td>
</tr>
<tr>
<td>System Interface</td>
<td>1.9</td>
<td>4.0</td>
<td>9.0</td>
</tr>
<tr>
<td>System Algorithm</td>
<td>1.9</td>
<td>3.8</td>
<td>9.8</td>
</tr>
<tr>
<td>Operational Scenario</td>
<td>6.4</td>
<td>13.6</td>
<td>26.3</td>
</tr>
</tbody>
</table>
How Reuse Is Addressed

Reuse operates in two directions [1]:

• Development with reuse (DWR): previously developed artifacts are reused on the current project
  – Addressed completely by the DWR partial development model

• Development for reuse (DFR): the current project is creating artifacts to be reused on other projects
  – One aspect of DFR development is that DFR costs more than ordinary development
    • Addressed by the DFR cost driver (covered there)
  – Another aspect of DFR is that the artifacts may be only partially completed, as during an IR&D project
    • Addressed by the DFR partial development model
Size Model – Partial Development Submodel

(Concepts here are simplified a little)

The basic DWR concept:
- If a reused SizeDriver is being brought in, that saves effort, and so we adjust the size by multiplying the raw size by a PartialDevFactor less than 1.
- The value of PartialDevFactor is based on the maturity of the reused SizeDriver, and is looked up in a table [24].
  - How fully developed was the SizeDriver?
- If there is no reuse for this SizeDriver, then PartialDevFactor = 1 (no adjustment).

<table>
<thead>
<tr>
<th>DWR Activity Level:</th>
<th>New</th>
<th>Design Modified</th>
<th>Design Implemented</th>
<th>Adapted for Integration</th>
<th>Adapted for Integration</th>
<th>Managed</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWR % for this AL through end</td>
<td>100.00%</td>
<td>83.00%</td>
<td>70.13%</td>
<td>56.88%</td>
<td>37.82%</td>
<td>17.50%</td>
</tr>
</tbody>
</table>

The basic development-for-reuse (DFR) concept is analogous:
- A product to be reused may be not be taken through the full development cycle (e.g., an IR&D project)
COSYSMO 3.0

Exponent Model

- Exponent model is expanded from Peña [4, 9]

\[ E = E_{\text{Base}} + SF_{\text{ROR}} + SF_{\text{PC}} + SF_{\text{RV}} \]

Where:
- \( E_{\text{Base}} \) = A minimum exponent for diseconomy of scale
- SF = scale factor
- \( ROR \) = Risk/Opportunity Resolution
- \( PC \) = Process Capability
- \( RV \) = Requirements Volatility

The effect of a large exponent is more pronounced on bigger projects
# Cost Drivers

- Here are the 13 COSYSMO 3.0 cost drivers:

<table>
<thead>
<tr>
<th>Driver Name</th>
<th>Data Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONOPS &amp; requirements understanding</td>
<td>Subjective assessment of the CONOPS &amp; the system requirements</td>
</tr>
<tr>
<td>Architecture understanding</td>
<td>Subjective assessment of the system architecture</td>
</tr>
<tr>
<td>Stakeholder team cohesion</td>
<td>Subjective assessment of all stakeholders</td>
</tr>
<tr>
<td>Level of service requirements</td>
<td>Subjective difficulty of satisfying the key performance parameters</td>
</tr>
<tr>
<td>Technology risk</td>
<td>Maturity, readiness, and obsolescence of technology</td>
</tr>
<tr>
<td># of Recursive levels in the design</td>
<td>Number of applicable levels of the Work Breakdown Structure</td>
</tr>
<tr>
<td>Development for reuse</td>
<td>Is this project developing artifacts for later reuse?</td>
</tr>
<tr>
<td># and Diversity of installations/platforms</td>
<td>Sites, installations, operating environment, and diverse platforms</td>
</tr>
<tr>
<td>Migration complexity</td>
<td>Influence of legacy system (if applicable)</td>
</tr>
<tr>
<td>Personnel/team capability</td>
<td>Subjective assessment of the team’s intellectual capability</td>
</tr>
<tr>
<td>Personnel experience/continuity</td>
<td>Subjective assessment of staff consistency</td>
</tr>
<tr>
<td>Multisite coordination</td>
<td>Location of stakeholders and coordination barriers</td>
</tr>
<tr>
<td>Tool support</td>
<td>Subjective assessment of SE tools</td>
</tr>
</tbody>
</table>

- Relative to COSYSMO 1.0 cost drivers, this model:
  - Drops Documentation
  - Adds Development for Reuse
  - Changes Process Capability to scale factor
The EMR (Effort Multiplier Ratio) of a cost driver is its maximum possible value divided by its minimum possible value; this is the impact of the cost driver.
Agenda:

• Preliminaries: motivation, history, research hypothesis, & methodology
• The priors; the COSYSMO 3.0 formula
• Closing: Developing the Final Model; future research
• Appendices: Bibliography; backup slides
Development of the Final Model

• I obtained a dataset of about 40 projects from an aerospace company
  – I fit a model to that via linear regression, using the Delphi results as Bayesian priors
    • Had to drop the planned Interoperability cost driver, due to not being addressed in the dataset
  – Also used recommendation to reduce impact of cost drivers as a Bayesian prior [11]
  – The result is the Final Model of COSYSMO 3.0

• Calibrating the model to the dataset and achieving $\text{PRED}(.30) \geq 50\%$ took some imagination
  – A simple-minded fit resulted in either:
    • Some non-credible parameter values; or
    • $\text{PRED}(.30) < 50\%$.
  – I was able to calibrate by:
    • Dropping a few outliers; and
    • Using a hill-climbing algorithm to find suitable parameter values.
Future Of COSYSMO 3.0

• Hope that existence of the model improves practice of systems engineering
  • Model is open—parameter values & definitions, estimating spreadsheet, etc are publically available
  • Model captures more project attributes than previous models
  • Easy to retrofit a database of existing projects
  • Hope that model enables production of a systems engineering cost estimate as standard practice
    • Developers, acquirers

• Future research topix:
  • Create a validated model for interoperability
    • Existing COSYSMO 3.0 work provides an excellent foundation
  • Create tailored models for different types of project
    • “Tailored” = some driver values are pre-filled in
    • Defense, software-intensive, ...
  • Estimating model for total development cost, based primarily on COSYSMO 3.0 drivers
    • Some work already done at Lockheed-Martin
  • Better integrate activity levels between DWR and DFR
  • Strengthen validation via additional datasets
    • 2nd dataset has PRED(.30) = 84%
Agenda:

• Preliminaries: motivation, history, research hypothesis, & methodology
• The priors; the COSYSMO 3.0 formula
• Closing: Developing the Final Model; future research
• Appendices: Bibliography; backup slides


Bibliography (4/5)


Backup Slides

- Impact of PCAP in COSYSMO 3.0
- Example parameter definition
- Numerical parameters of the Final Model
- Placing Process Capability: cost driver or scale factor?
- Continuity and the Rosetta Stone
- Coordination with COCOMO III
Effect of Different PCAP Ratings (VL/N/EH) on Projects of Different Sizes
Showing Effort Ratios to VL Rated Project
Example Cost Driver: Dev. For Reuse

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:

<table>
<thead>
<tr>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>No reuse at all.</td>
<td>Artifacts will be reused only on</td>
<td>Artifacts will be reused across</td>
<td>Artifacts will be reused across</td>
<td>Artifacts will be reused across</td>
</tr>
<tr>
<td></td>
<td>the current project.</td>
<td>across the program.</td>
<td>a product line.</td>
<td>multiple product lines.</td>
</tr>
</tbody>
</table>
## Cost Driver and Scale Factor Ratings

<table>
<thead>
<tr>
<th>Cost Driver</th>
<th>Step Size</th>
<th>Effort Multipliers</th>
<th>Scale Factor</th>
<th>Scale Factor Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Text Rating:</strong></td>
<td><strong>Very Low</strong></td>
<td><strong>Low</strong></td>
<td><strong>Nominal</strong></td>
<td><strong>High</strong></td>
</tr>
<tr>
<td><strong>Numeric Rating:</strong></td>
<td>-2.0</td>
<td>-1.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>CONOPS &amp; Requirements Understanding</td>
<td>0.765</td>
<td>1.71</td>
<td>1.31</td>
<td>1.00</td>
</tr>
<tr>
<td>Architecture Understanding</td>
<td>0.805</td>
<td>1.54</td>
<td>1.24</td>
<td>1.00</td>
</tr>
<tr>
<td>Stakeholder Team Cohesion</td>
<td>0.802</td>
<td>1.55</td>
<td>1.25</td>
<td>1.00</td>
</tr>
<tr>
<td>Level of Service Requirements</td>
<td>1.277</td>
<td>0.61</td>
<td>0.78</td>
<td>1.00</td>
</tr>
<tr>
<td>Technology Risk</td>
<td>1.262</td>
<td>0.63</td>
<td>0.79</td>
<td>1.00</td>
</tr>
<tr>
<td># of Recursive Levels in the Design</td>
<td>1.179</td>
<td>0.72</td>
<td>0.85</td>
<td>1.00</td>
</tr>
<tr>
<td># and Diversity of Installations/Platforms</td>
<td>1.238</td>
<td>(Invalid)</td>
<td>1.00</td>
<td>1.24</td>
</tr>
<tr>
<td>Migration Complexity</td>
<td>1.252</td>
<td>(Invalid)</td>
<td>1.00</td>
<td>1.25</td>
</tr>
<tr>
<td>Personnel/Team Capability</td>
<td>0.831</td>
<td>1.45</td>
<td>1.20</td>
<td>1.00</td>
</tr>
<tr>
<td>Personnel Experience/Continuity</td>
<td>0.858</td>
<td>1.36</td>
<td>1.17</td>
<td>1.00</td>
</tr>
<tr>
<td>Multisite Coordination</td>
<td>0.812</td>
<td>1.52</td>
<td>1.23</td>
<td>1.00</td>
</tr>
<tr>
<td>Tool Support</td>
<td>0.892</td>
<td>1.26</td>
<td>1.12</td>
<td>1.00</td>
</tr>
<tr>
<td><strong>Risk &amp; Opportunity Management</strong></td>
<td>-0.0120</td>
<td>0.0602</td>
<td>0.0482</td>
<td>0.0361</td>
</tr>
<tr>
<td><strong>Process Capability</strong></td>
<td>-0.0107</td>
<td>0.0536</td>
<td>0.0429</td>
<td>0.0322</td>
</tr>
<tr>
<td><strong>Requirements Volatility</strong></td>
<td>0.0095</td>
<td>0.0000</td>
<td>0.0095</td>
<td>0.0189</td>
</tr>
</tbody>
</table>
# COSYSMO 3.0 Final Model Constants

<table>
<thead>
<tr>
<th></th>
<th>Productivity Factor</th>
<th>26.33</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBase</td>
<td>Exponent Base</td>
<td>1.0332</td>
</tr>
</tbody>
</table>
Placement of Process Capability (1/2)

- Process Capability (PROC) was a cost driver in COSYSMO 1.0, but there were arguments that it should be a scale factor instead.
- With an earlier version of the model, I generated this table:

<table>
<thead>
<tr>
<th></th>
<th>PROC as CD</th>
<th>PROC as SF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Driver Fit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard Error of Residuals</td>
<td>0.4829</td>
<td>0.4994</td>
</tr>
<tr>
<td>R squared</td>
<td>0.9890</td>
<td>0.9882</td>
</tr>
<tr>
<td>F-statistic</td>
<td>570.80</td>
<td>571.20</td>
</tr>
<tr>
<td>Scale Factor Fit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard Error of Residuals</td>
<td>0.3985</td>
<td>0.1930</td>
</tr>
<tr>
<td>R squared</td>
<td>0.9911</td>
<td>0.9979</td>
</tr>
<tr>
<td>F-statistic</td>
<td>3946</td>
<td>11280</td>
</tr>
</tbody>
</table>

- One argument in favor of “scale factor” is that its cost driver fit is only slightly worse, but its scale factor fit is much better.
Placement of Process Capability (2/2)

• A second argument is that members of the Working Group have the intuition that “scale factor” is likely correct, as a poor process would have a proportionally greater impact on a larger project.

• A third argument is that “scale factor” would agree with its placement in COCOMO II (and, apparently, COCOMO III).
Definition Modifications & the Rosetta Stone

- Users of previous versions of COSYSMO want to carry forward as much of their estimation database as possible (“continuity”). So support is provided via a Rosetta Stone document which provides instructions on how to re-rate existing drivers under COSYSMO 3.0.

<table>
<thead>
<tr>
<th>Degree of Definition Change</th>
<th>Rosetta Stone Instruction for How To Change Old Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>No change</td>
<td>“No change”</td>
</tr>
<tr>
<td>Moderate change</td>
<td>“No change”</td>
</tr>
<tr>
<td>Substantial change</td>
<td>“Decrement old rating by x steps”</td>
</tr>
</tbody>
</table>
# Excerpts from the Rosetta Stone

**Element Type Key:**
- **CD** = Cost Driver
- **SF** = Scale Factor

**Color (Degree of Change) Key:**
- No Change in Definition
- No Change in Rating
- Rating Change
- New or Deleted

<table>
<thead>
<tr>
<th>Element from COSYSMO 1.0 (or other model as shown)</th>
<th>COSYSMO 3.0 Element</th>
<th>Instructions for 3.0 Rating</th>
<th>Justification (for Working Group members)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD: Requirements Understanding</td>
<td>CD: CONOPS and Requirements Understanding</td>
<td>Same rating.</td>
<td>1.0 CD is now a viewpoint in a larger 3.0 CD; the 1.0 rating should be carried forward in that context, without any mis-rating.</td>
</tr>
<tr>
<td>CD: Documentation Match to Life Cycle Needs</td>
<td>Dropped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD: # and Diversity of Installations/Platforms</td>
<td>(same)</td>
<td>Same rating.</td>
<td>Same definition.</td>
</tr>
<tr>
<td>CD: Personnel/Team Capability</td>
<td>(same)</td>
<td>Same rating.</td>
<td>Minor text definition improvement should allow the 1.0 rating to carry forward to a 3.0 rating.</td>
</tr>
<tr>
<td>CD: Process Capability</td>
<td>(same)</td>
<td>Same rating.</td>
<td>Same definition.</td>
</tr>
<tr>
<td>CD: Personnel Experience/Continuity</td>
<td>(same)</td>
<td>Decrement old rating by half a level.</td>
<td>One of two viewpoints has been &quot;shifted up&quot; one level; e.g., 3 years’ experience was Nominal, but in 3.0 it’s Low.</td>
</tr>
<tr>
<td>CD: Multisite Coordination</td>
<td>(same)</td>
<td>Same rating.</td>
<td>Same definition.</td>
</tr>
<tr>
<td>CD: Tool Support</td>
<td>(same)</td>
<td>Decrement old rating by a level.</td>
<td>The rating scale has been &quot;shifted up&quot; one level; e.g., &quot;Strong, mature tools&quot; was rated High, but in 3.0 it’s Nominal. In addition, the topic of life cycle coverage was added to the rating scale. Decrementation by a level should not result in a significant mis-rating.</td>
</tr>
</tbody>
</table>
The Final Model has been coordinated with Brad Clark’s in-progress COCOMO III definition effort, with these results:

- Essentially identical definitions of Risk/Opportunity Management scale factor.
- Essentially identical definitions of Multi-Site Development cost driver.
- COSYSMO 3.0 Development for Reuse cost driver taken from COCOMO II.
- COSYSMO 3.0 Personnel/Team Capability cost driver definition modified to agree with COCOMO II’s.
Coordination with COCOMO III (2/2)

- A 2012 paper* was published distinguishing the scopes of COCOMO and COSYSMO in a project; Brad and the COSYSMO 3.0 Working Group coordinated on an (unpublished) update ("COCOMO – COSYSMO Estimation Boundaries")