COSATMO:
Developing The Harmonized COSYSMO 3.0 Model Including the Coalesced Effort Multiplier Submodel

Jim Alstad
USC Center for Systems and Software Engineering

2015 Ground Station Architecture Workshop
Working Group 11D – COSYSMO 3.0
13:00 March 4, 2015

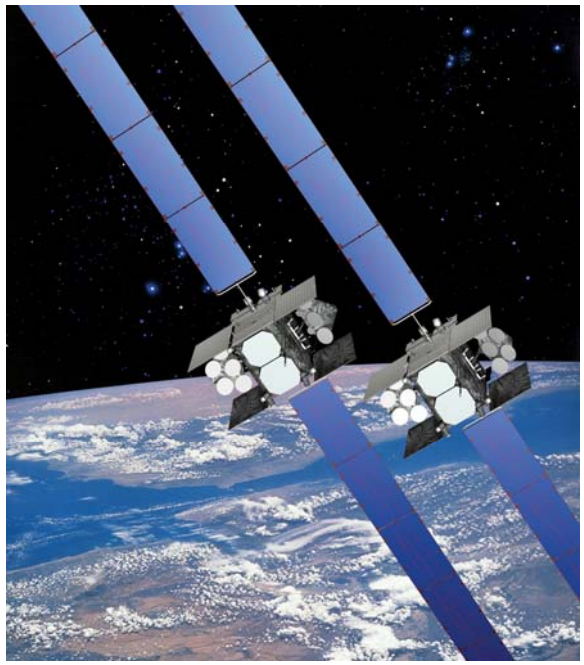
© 2015 by University of Southern California.
Published by The Aerospace Corporation with permission.

Agenda

Agenda:

- **COSATMO and COSYSMO**
  - **Summarized results of 2014 Working Group**
- **Presentation of Harmonized/CEM COSYSMO 3.0 model**
- **Bibliography**

The Problem



- How much will the total system cost?
- Is one phase being optimized while increasing total cost?
- Is the system affordable?
- Does the acquisition comply with the Better Buying Power intiatives (DoD)?

COSATMO Objective

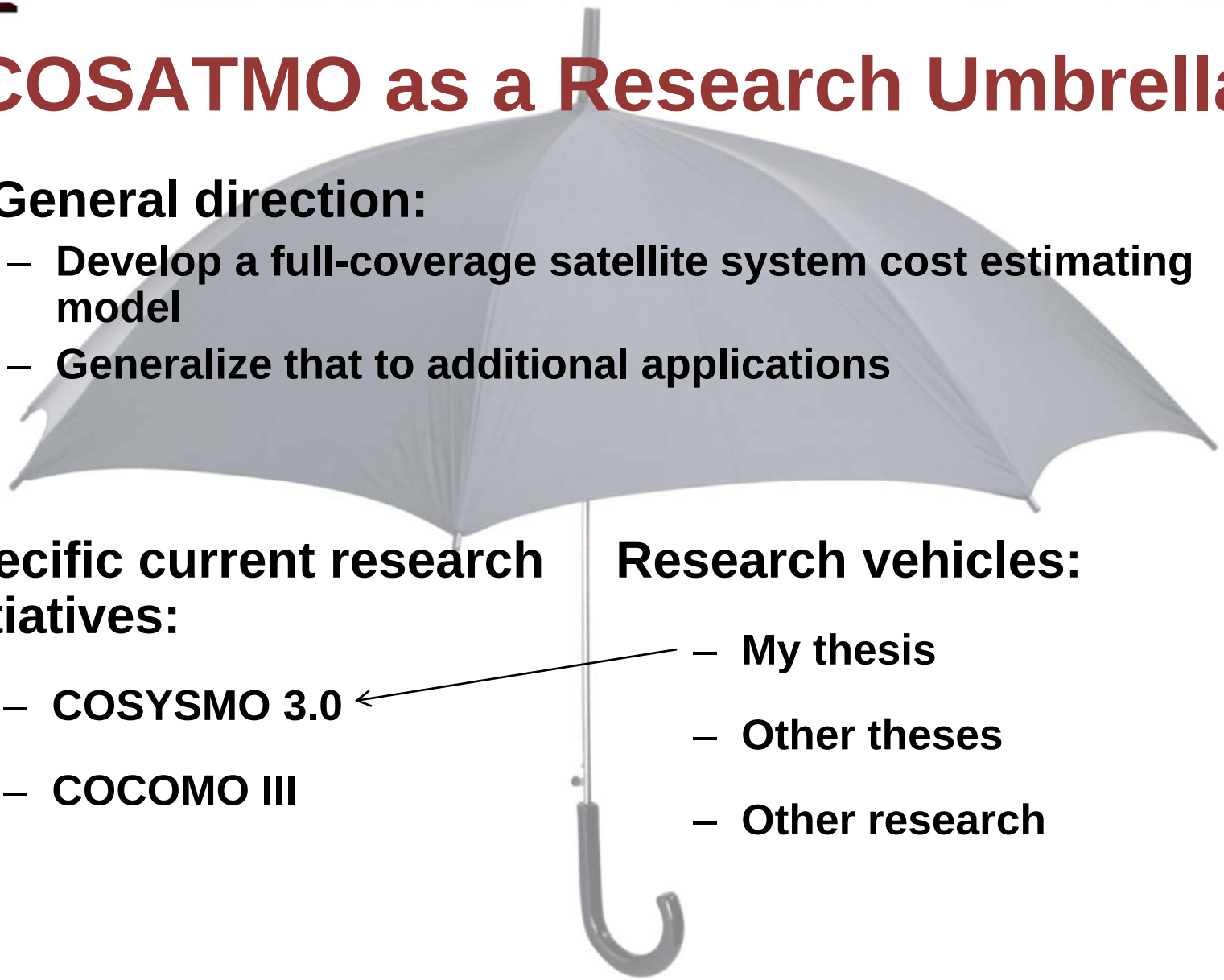
- **Context:**
  - **Current and future trends create challenges for full-system cost estimation**
    - **Emergent requirements, rapid change, net-centric systems of systems, COTS, clouds, apps, widgets, high assurance with agility, multi-mission systems**
  - **Current development practices can minimize cost of one phase, such as development, while raising full-system cost**
- **The COSATMO project is developing a modern full-system cost model (first space systems, then other DoD domains)**
  - **"Constructive SATellite cost MOdel"**
  - **Current estimating models focus on one aspect, such as system engineering**
  - **COSATMO will enable:**
    - **System-level trades to be handled within a single model**
    - **Easy customer evaluation of full-system cost**
    - **Modern technologies to be covered**

Summarized Results from 2014 Working Group

COSATMO submodels covering ground station cost with most important cost drivers ("(s)" indicates size-like):

- **Initial development of software:**
  - **(Probably already covered by COCOMO II, COCOTS, COPLIMO)**
- **Initial development of prime mission equipment:**
  - **Amount of new development required, # PME sites (s), required site availability & reliability**
- **Initial development of (other) facilities**
  - **# facilities (s), location of facilities (especially US vs foreign)**
- **Operations & support**
  - **# years of operation (s), # FTE staff (with labor mix) (s)**
- **System-level**
  - **Accreditation (IA, etc), required security (especially cross-domain)**

COSATMO as a Research Umbrella



- **General direction:**
  - **Develop a full-coverage satellite system cost estimating model**
  - **Generalize that to additional applications**

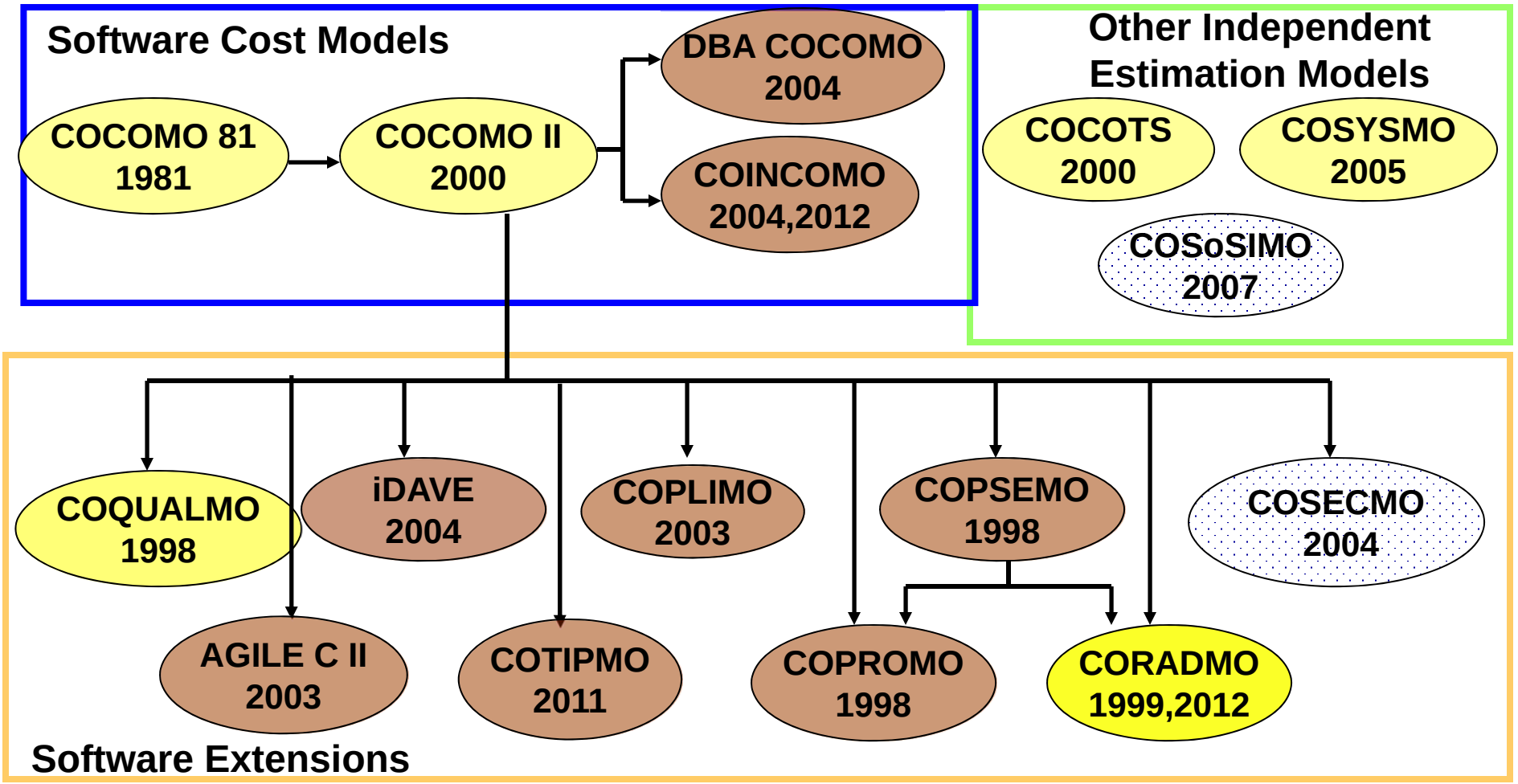
Specific current research initiatives:

- **COSYSMO 3.0**
- **COCOMO III**

Research vehicles:

- **My thesis**
- **Other theses**
- **Other research**

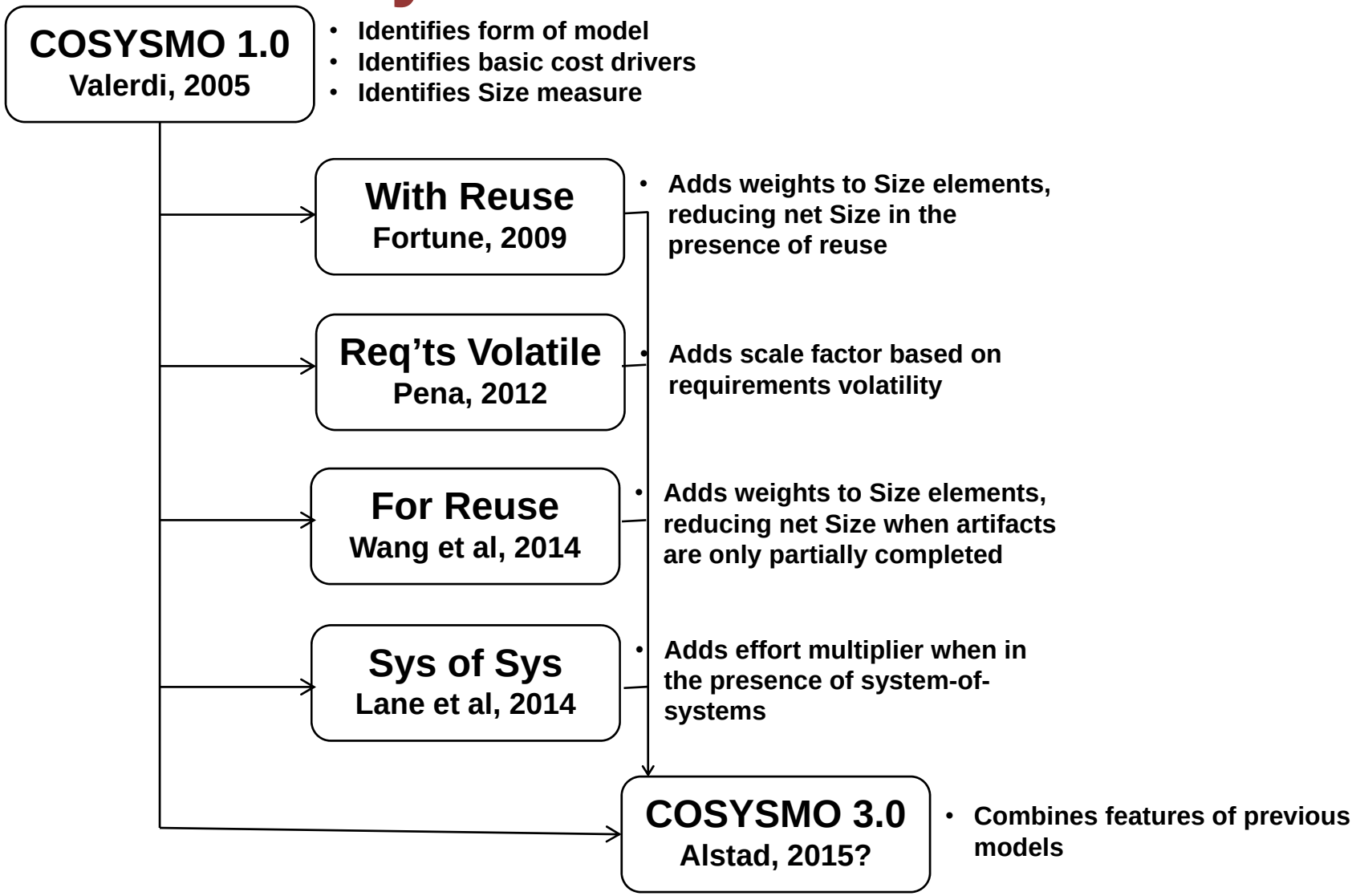
COCOMO Family of Cost Models



Legend:
Model has been calibrated with historical project data and expert (Delphi) data
Model is derived from COCOMO II
Model has been calibrated with expert (Delphi) data

Dates indicate the time that the first paper was published for the model

History of COSYSMO Models



COSYSMO 3.0 Directions (Adapted from ARR slides [8])

Harmonize existing COSYSMO family models:

- **Several factors affecting the COSYSMO cost model have been shown to be valuable in increasing estimation accuracy (terminology from [5]):**
  - **Reuse (simple model--SEWR) [3]**
  - **Reuse (with SEFR) [1]**
  - **Requirements volatility (SERV) [4]**

  The rating scales for these could be integrated into a comprehensive COSYSMO model.

Enhancement planned for inclusion:

- **System-of-system considerations are hypothesized to affect system engineering costs:**
  - **Interoperability considerations [6]**
- **Reduce the number of Effort Multipliers from 14 – "Coalesced Effort Multipliers"**

COSYSMO 3.0 Directions Part 2

Enhancements under discussion:

- **Explore a model for total development cost based primarily on the COSYSMO parameters (Cole)**

Harmonized/CEM COSYSMO 3.0 Status

What will be covered today is an early prototype of the Harmonized/CEM COSYSMO 3.0 model

- **Fairly complete definitions available (details not in these slides included in estimating spreadsheet)**
- **Some estimating parameters have been determined via a Delphi exercise**
  - **Those will be refined today**
  - **Thereafter further parameter refinement will be needed**
- **Before today, no projects have been estimated using the model**

Harmonized/CEM COSYSMO 3.0 Top-Level Model

$$PM_{C3} = A_{C3} \cdot (Size_{C3})^{E_{C3}} \cdot \prod_{j=1}^{5} EM_{C3,\,j}$$

Elements of the Harmonized/CEM model:

- ***Calibration parameter A***
- **Interoperability**
- **Size model**
  - **eReq submodel**
  - **Partial development submodel**
- **Exponent (E) model**
  - **Submodels for 3 scale factors**
- **Effort multipliers EM**
  - **Revised from earlier models**
- **Multi-subproject model**

Harmonized COSYSMO 3.0 Interoperability Model

- Lane & Valerdi [6] propose that interoperability be considered a cost influence in the COSYSMO family
- Motivation: if a system is part of a system-of-systems, then that context is reflected in interoperability requirements on the system
- Two ways this influence could be manifested are proposed:
  - Method 1: Add a new effort multiplier
  - Method 2: Adjust the easy/medium/difficult rating scale for system interfaces (part of the Size model)
- Both Methods are shown in this presentation; presumably only one would be retained in COSYSMO 3.0.

Harmonized COSYSMO 3.0 Size Model

$$Size_{C3} = \sum_{Prods} eReq(Type(Prod), Difficulty(Prod)) \cdot$$

$$PartialDevFactor(Phase_{Start}(Prod), Phase_{End}(Prod))$$

- **Prod is one of the four system engineering products that determines size in COSYSMO family (per [2]):**
  - **System requirement**
  - **System interface**
  - **System algorithm**
  - **Operational scenario**
- **For simplicity in model explanation, each individual Prod is considered separately**
- **There are two submodels:**
  - **Equivalent nominal requirements ("eReq")**
  - **Partial development**

Size Model – eReq Submodel (1/2)

- The eReq submodel is unchanged from [2].
  - Though terminology is a little different
  - Also, see next slide
- The submodel computes the size of a Prod, in units of eReq ("equivalent nominal requirements")
- Each Prod is evaluated as being easy, nominal, or difficult.
- Each Prod is looked up in this size table to get its number of eReq:

Prod Type	Easy	Nominal	Difficult
System Requirement	0.5	1.0	5.0
System Interface	1.1	2.8	6.3
System Algorithm	2.2	4.1	11.5
Operational Scenario	6.2	14.4	30.0

Size Model – eReq Submodel (2/2)

Adjustment for interoperability (Method 2):

- **[6] proposes (in its Table 3) that the table that defines the easy/medium/hard rating scale for a system interface (from [2]) be adjusted by adding a new row (the last row in this table):**

Easy	Medium	Difficult
Simple messages and protocols	Moderate communication complexity	Complex protocol(s)
Uncoupled	Loosely coupled	Tightly coupled
Strong consensus among stakeholders	Moderate consensus among stakeholders	Low consensus among stakeholders
Well behaved	Predictable behavior	Emergent behavior
Domain or enterprise standards employed	**Functional standards employed**	**Isolated or connected systems with few or no standards**

Size Model – Partial Development Submodel (1/2)

- The following new partial development submodel is proposed
  - It is a generalization of the Generalized Reuse Framework model [1] (Generalized^2 Reuse Framework?)
  - An alternate submodel is discussed below

$$PartialDevFactor(Phase_{Start}(Prod), Phase_{End}(Prod))$$

- A Partial Development Factor from 0.0 to 1.0 is assigned based on the starting and ending phase of development on Prod
  - The phases: Requirements and Architecture, Detailed Design, Implementation, Integration & Test, Operational Test & Evaluation
  - A Prod may enter late, typically because it's being reused
  - A Prod may exit early, typically because this is an IR&D project
  - ***PartialDevFactor*** = 1.0 for a complete development life cycle

Size Model – Partial Development Submodel (2/2)

- ***PartialDevFactor*(*Phase*$_{Start}$, *Phase*$_{End}$) might be approximated by:**

$$\sum_{p=Phase_{Start}}^{Phase_{End}} Fraction\ of\ development\ cost\ for\ phase\ p$$

  - **But there might be some overhead for early entry or early exit**

Alternate *PartialDevFactor* submodel:

- **Use results from Generalized Reuse Model [1]**
  - **Restricted to Prods that either start at the beginning or finish at the end**
- **Would be employed if data didn't validate the proposed model**

Harmonized COSYSMO 3.0 Exponent Model

- **Exponent model is unchanged from Peña [4, 9]**

$$E_{C3} = E_{COSYSMO} + SF_{RV} + SF_{RESL} + SF_{PM}$$

- ***E*$_{COSYSMO}$ = 0.9995 [2]**
  - **Was 1.06 in COSYSMO 1.0 [2]**
  - **Adjusted so that Nominal values of SFs yield 1.06.**
- ***SF*$_{RV}$ explained on next 3 slides**
- **Table of all SF values given thereafter**

Adapted from slide 11 of [9]

Exponent – SF Submodel

$$SF_{RV} = \left(C \cdot \left(\frac{REVL}{100} \right) \cdot w_{vl} \right)$$

Expected *REVL* is rated as Very Low, Low, Moderate, High, and Very High

- **Where**
  - $REVL$ **= The % of the baseline requirements that is expected to change over the system lifecycle – see next slide**
  - C = **Scale factor constant = 0.05 (calibration parameter)**
  - w_{vl} **= aggregate lifecycle phase volatility weighting factor**

$$w_{v1} = \left(w_{concep}\Theta_{concep} + w_{dev}\Theta_{dev} + w_{test}\Theta_{test} + w_{tran_ops}\Theta_{tran_ops} \right)$$

- **And:**
  - w_l **= weighting factor for each life cycle phase[1] (determined by experts [4])**
  - Θ_l **= % of total requirements changes per life cycle phase (taken from data from 25 projects [4])**
  - l **= life cycle phases**

[1]*Life Cycle Phases: Conceptualize, Development, Operational Test and Evaluation, and Transition to Operation*

Adapted from slide 14 of [9]

Exponent – REVL Submodel – Ranking

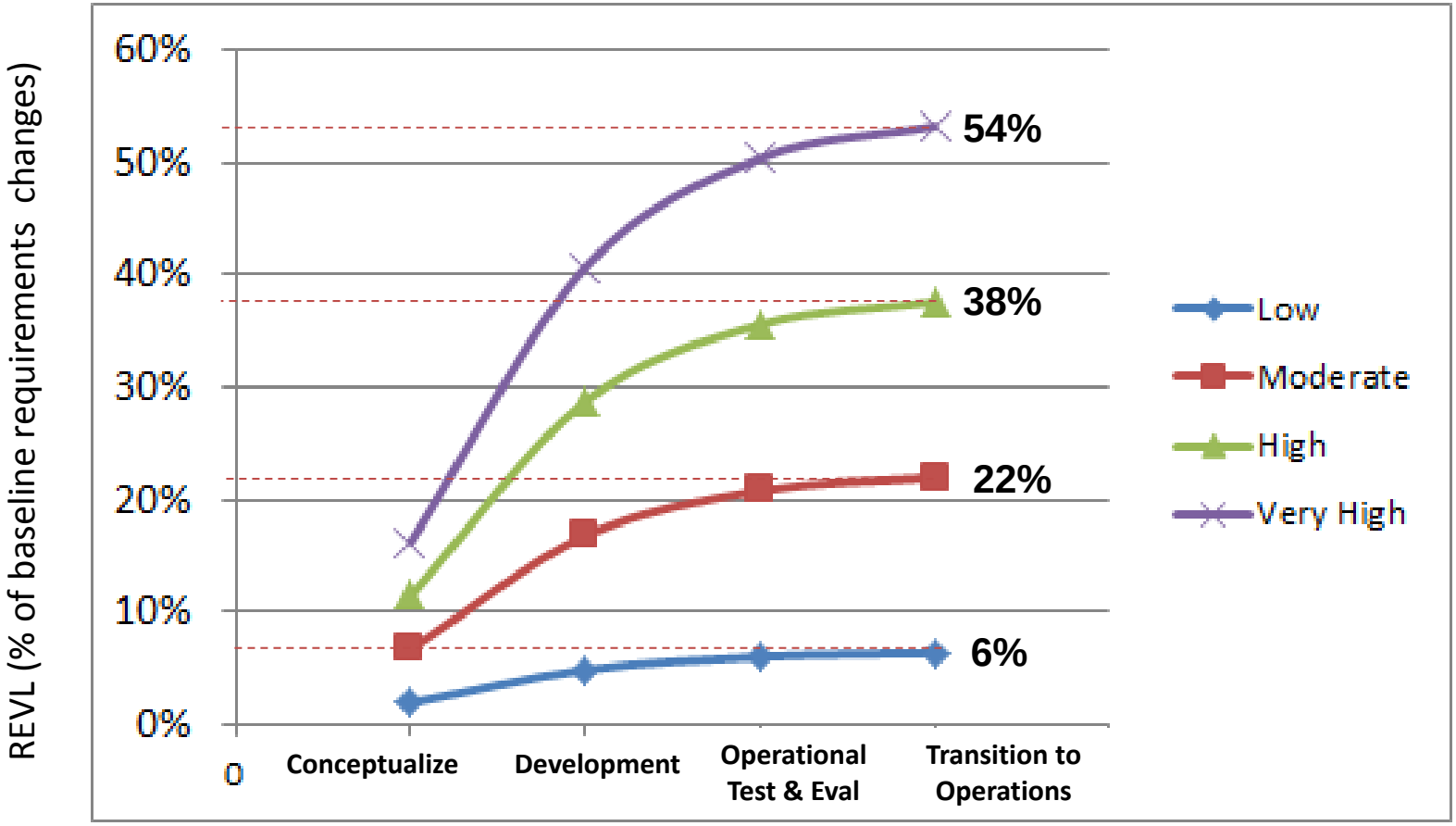
Have been modified a little in Harmonized COSYSMO 3.0.

Characteristic / Rating	Very Low	Low	Moderate	High	Very High	Weight
	< 1.5	>1.5-2.5	>2.5-3.5	>3.5-4.5	> 4.5	
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%

Developed based on surveys of experienced S/W and Systems Engineers (N =38) [4]

Adapted from slide 29 of [9]

Exponent – REVL – Ranking to Percentage



- **Based on data from 25 projects [4]**

Table of Scale Factor Values

Defi-nition	Abbrev	Extra Low	Very Low	Low	Nom-inal	High	Very High	Extra High
Req'ts Vola-tilty	SF_{RV}		.00000	.00300	.01100	.01900	.02700	
Risk & Opp. Reso-lution	SF_{RESL}		.05333	.04267	.03200	.02133	.01067	.00000
Pro-cess Matur-ity	SF_{PM}	.03500	.02917	.02333	.01750	.01167	.00583	.00000

Harmonized/CEM COSYSMO 3.0 Effort Multiplier Model

- Effort multiplier rating scales are organized in three levels:
  - 1. Effort Multiplier
  - 2. Effort Multiplier Element
  - 3. Viewpoint
    - May or may not be present
    - In a few cases, has sub-viewpoints
- How to integrate ratings:
  - Subjective weighting is used to integrate Viewpoint ratings into an Effort Multiplier Element rating
  - Weighted summing is used to integrate Effort Multiplier Element ratings into an Effort Multiplier rating
    - May not be an exact rating (e.g., may be 70% of the way from Nominal to High)
  - An Effort Multiplier rating is converted into a numeric effort multiplier value.

Harmonized/CEM COSYSMO 3.0 EMs & EMEs (1/2)

Effort Multiplier	Effort Multiplier Elements
Problem Understanding	Understanding of CONOPS
	Understanding of Functional Requirements
	Understanding of Non-Functional Requirements
Solution Understanding and Difficulty	Architecture Understanding
	NDI Understanding
	System-Critical Technology Understanding
	System-Critical Technology Maturity
	System Definition Difficulty
	# of Recursive Levels in the Design
Personnel/Team Capability	Team/Stakeholder Cohesion
	Team/Stakeholder Stabiity (rated by annual turnover)

Harmonized/CEM COSYSMO 3.0 EMs & EMEs (2/2)

Effort Multiplier	Effort Multiplier Elements
Productivity Aids	Capability & Experience with Toolset
	Toolset Coverage, Maturity, & Integration
	Collaboration Support (rating levels above Nominal involve increasing levels of automated collaboration support such as virtual presence, social networking technology, data services, and knowledge-based negotiation convergence aids)
	Collocation
Performance Constraints	Legacy Migration
	Platform Diversity
	Available Schedule (% of nominal schedule)
	Interoperability
	SEFR (System Engineering for Reuse)

Harmonized/CEM COSYSMO 3.0 Interoperability Approach via EM

Adjustment for interoperability (Method 1):

- **"Interoperability" should perhaps be included in the Effort Multiplier hierarchy**
- **Table 2 of [6] proposes this rating scale, depending on whether the project is for an existing system or a new system:**

Type of Development	Level				
	Very Low	Low	Nominal	High	Very High
Existing systems (based upon LISI levels)	Isolated	Con-nected	Functional standards employed	Domain standards employed	Enterprise standards employed
New system (s) (based upon LCIM conceptual levels)	System-specific data	Docu-mented data	Aligned static data	Aligned dynamic data	Harmon-ized data

Harmonized/CEM COSYSMO 3.0 Rating Understanding EMEs (1/5)

Some of the EMEs include the term "Understanding":

- **Understanding of CONOPS**
- **Understanding of Functional Requirements**
- **Understanding of Non-Functional Requirements**
- **Architecture Understanding**
- **NDI Understanding**
- **System-Critical Technology Understanding**

The CEM model introduces a new, uniform approach for rating Understanding EMEs

- **Explained in this group of slides**

Harmonized/CEM COSYSMO 3.0 Rating Understanding EMEs (2/5)

1. **Identify & assign "factors" relevant to the project**
   a. **Identify the project's factors**
      - **A factor is any aspect of the project that has a major influence on the project's technologies or general approach; especially aspects that are new to the team. Some examples:**
        - **Government or commercial customer?**
        - **Cyber? Physical? Human? (one or more)**
        - **System's organizational relationships**
        - **Other aspects having a major influence on the system or project**
   b. **Assign each factor to the Understanding EMEs**
      - **Assign a factor to multiple EMEs when appropriate**
      - **Result: Each EME has (usually) one or more factors assigned to it.**

Harmonized/CEM COSYSMO 3.0 Rating Understanding EMEs (3/5)

2. For each EME, rate the project's understanding on 3 dimensions
   a. Absorb the 6-level understanding scale ("Bloom's taxonomy") (level definitions on rating spreadsheet):
      - Knowledge
      - Comprehension
      - Application
      - Analysis
      - Evaluation
      - Synthesis
   b. Rate each EME based on the development team's understanding of the EME's factors on these 3 dimensions using the table below (and in the spreadsheet):
      - Best Understanding Level in the Team
      - Average Understanding Level in the Team
      - Factor Bridging Level
        - How well do individuals know multiple factors?

Harmonized/CEM COSYSMO 3.0 Rating Understanding EMEs (4/5)

	Dimensions of Understanding for an EME's Factors		
Rating Level	**1. Best Understanding Level on Team**	**2. Team's Average Understanding Level**	**3. Factor Bridging Level**
Extra Low	Basic Knowledge of each factor covered	Minimal Knowledge of most factors	Some factor pairs at Knowledge level
Very Low	Comprehension, some Application of most factors covered	Knowledge of most factors, some Comprehension and Application of some factors	Some pairs at Comprehension level
Low	Application, some Analysis of most factors covered	Comprehension and Application of most factors, some Analysis of some factors	Some multi-factor bridging at Comprehension, Application levels
Nominal	Analysis, some Evaluation and Synthesis of some factors covered	Comprehension and Application of each factor, some Analysis of most factors	Some multi-factor bridging at Analysis level
High	Analysis, some Evaluation and Synthesis of most factors covered	Analysis of most factors, some Evaluation and Synthesis of some factors	Most factors bridged at Analysis level. Some at Evaluation level
Very High	Analysis, most Evaluation and Synthesis of most factors covered	Analysis of each factor, some Evaluation and Synthesis of most factors	Most factors bridged at Evaluation and Synthesis levels
Extra High	Evaluation and Synthesis of each factor covered	Most Evaluation and Synthesis of most factors	All factors bridged at Evaluation and Synthesis levels

Harmonized/CEM COSYSMO 3.0 Rating Understanding EMEs (5/5)

2. **For each EME, rate the project's understanding on 3 dimensions**
   c. **Spreadsheet will combine the three ratings (via weights) into an overall rating for the Understanding EME**

Harmonized/CEM COSYSMO 3.0 Multi-Subproject Model

- **Sometimes a project consists of multiple subprojects**
  - **Where the subprojects use significantly different effort multipliers.**
  - **However, scale factors should apply to the project as a whole.**
- **Example:**
  - **Part of a project is SEFR; the rest is not**
- **The equation below is adapted from Equation 2 of [10] and is based on the Multiple Modules model of COCOMO II**
  - **When applicable, it supersedes the Top-Level Model**

$$PM_{C3M} = A_{C3} \cdot (Total\,Size_{C3})^{E_{C3}} \cdot \sum_{i \in Subprojects} \left(\frac{Part_i\,Size_{C3}}{Total\,Size_{C3}} \cdot \prod_{j=1}^{5} EM_{C3:i,j} \right)$$

Bibliography (1/2)

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

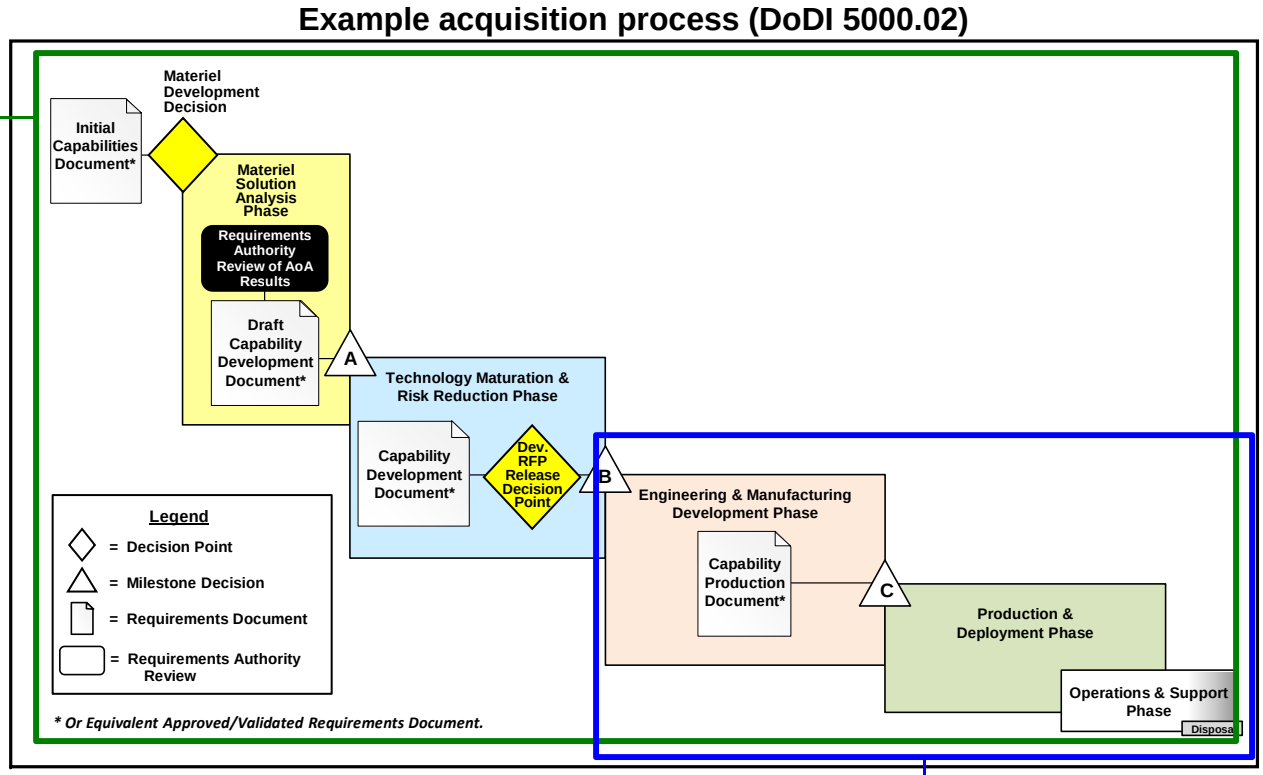
Bibliography (2/2)

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.

Backup Charts

The Solution

COSATMO assists acquirers and developers during these phases (highest payoff during early phases)



COSATMO estimates the cost for these phases

COSATMO Concept

- Focused on current and future satellite systems
  - Accommodating rapid change, evolutionary development, Net-Centric SoSs, Families of systems, DI2E SWASe's
    - Software, Widgets, Assets, Services, etc.
  - Recognizes new draft DoDI 5000.02 process models
    - Hardware-intensive, DoD-unique SW-intensive, Incremental SW-intensive, Accelerated acquisition, 2 Hybrids (HW-, SW-dominant)
  - Supports affordability analyses (total cost of ownership):
    - Covers full life cycle: definition, development, production, operations, support, phaseout
    - Covers full system: satellite(s), ground systems, launch
    - Covers hardware, software, personnel costs
- Extensions to cover systems of systems, families of systems
- Several PhD dissertations involved (as with COSYSMO)
  - Incrementally developed based on priority, data availability

Approach

- **Technical approach:**
  - **Develop a satellite system cost model**
    - **Divide overall system cost into segments. For each segment:**
      - **Identify an existing cost model (one or more) that covers it, or**
      - **Develop a new cost model for the segment**
    - **For any new cost models, follow the well-developed COCOMO-family methodology:**
      - **Identify cost drivers**
      - **Obtain expert opinion on impact of cost driver**
      - **Combine that statistically with cost data from actual systems**
      - **Iterate as needed**
  - **Generalize to other DoD systems**
- **The near-term activities, then, are:**
  - **Convene groups of experts to identify cost drivers and impacts**
  - **Identify sources of data**

Near-Term Work Approach

- **Developing a segment model typically consists of two topics (which are somewhat independent):**
  1. **Identifying cost drivers and determining which are most important (compare slides 9-11)**
  2. **Gathering actual, total segment costs for multiple systems, including actual values of cost driver**
  - **After 1 & 2 are complete, data can be analyzed and the segment cost model can be finalized**
- **Segments (see slides 7-8) that seem to have the highest benefit/cost ratio for near-term work on either or both topics:**
  - **Total engineering cost (all through EMD phase—slides 3, 7)**
  - **Operation & support**
  - **Other ground segments**

Segments of Satellite System Cost

- **Total satellite system cost [tied to slide 3 phases] =**
  - **System engineering cost [EMD]**
  - **+ Satellite software cost [EMD]**
  - **+ Satellite vehicle hardware development [EMD] and production [Prod] cost**
  - **+ Launch cost [Deploy]**
  - **+ Initial ground software cost [EMD]**
  - **+ Initial ground custom equipment cost [EMD]**
  - **+ Initial ground facility (buildings, communications, computers, COTS software) cost [EMD]**
  - **+ Operation & support cost [Deploy, O&S]**
- **Updated at GSAW (Feb 2014)**
- **Model as sum of submodels is new structure in COCOMO family**

COSATMO Segment Tentative Models

- System engineering: COSYSMO, perhaps with add-ons
- Satellite vehicle hardware development and production: Current Aerospace hardware cost model(s); exploring extensions of COSYSMO for hardware cost estimation
- Satellite vehicle, ground system software development: COCOMO II, COCOTS, perhaps with add-ons
- Launch model: similarity model, based on vehicle mass, size, orbit
- Ground system equipment, supplies: construction, unit-cost, services cost models
- Operation & support: labor-grade-based cost models, software maintenance models

Key Overall Satellite System Cost Drivers

- **Most Important:**
  - **Complexity, Architecture Understanding, Mass, Payload TRL level/Technology Risk, and Requirements Understanding.**
- **Important:**
  - **Reliability, Pointing Accuracy, Number of Deployables, Number of Key Sponsors, Data Rate, and Security Requirements for Communications.**
- **Determined at COCOMO Forum (Oct 2013)**

Ground System Segment Development (1/2)

- Determined at GSAW (Feb 2014)
- Ground system-wide cost drivers
  - Most important: Accreditation (information assurance, etc), Required security
  - Also important: # satellites*
- Initial software cost drivers
  - Required data throughput
  - Generally handled by COCOMO II, COCOTS, COPLIMO

*Indicates a size measure

Ground System Segment Development (2/2)

- Ground custom equipment cost drivers
  - Most important: Amount of new development required, # of custom equipment sites*, Required site availability & reliability, Required site security
  - Also important: # driving requirements*
- Ground facility cost drivers
  - Most important: # facilities*, location of facilities (especially US vs foreign), # ground RF terminals*
  - Also important: Facility "reuse"
- Operation and support cost drivers
  - Most important: # years of operation*, # FTE staff (with labor mix)*
  - Also important: Size of software maintained*, Leased line cost*, level of automation

*Indicates a size measure

Medium-Term Issues

Model-changing issues:

1. **Use of small satellites vs more traditional satellites vs mixed**
2. **Ownership model (own vs leased services, etc)**
3. **Is support for multiple missions required?**

Develop a phased cost model.

Is this a reasonable generalization to other domains:

- **Total system cost =**
  - **System engineering cost**
  - **+ Embedded software cost**
  - **+ Hardware development cost through first article**
  - **+ Deployment cost**
  - **+ Initial logistics software cost**
  - **+ Initial logistics custom equipment cost**
  - **+ Initial logistics facility cost**
  - **+ Operation & support cost?**

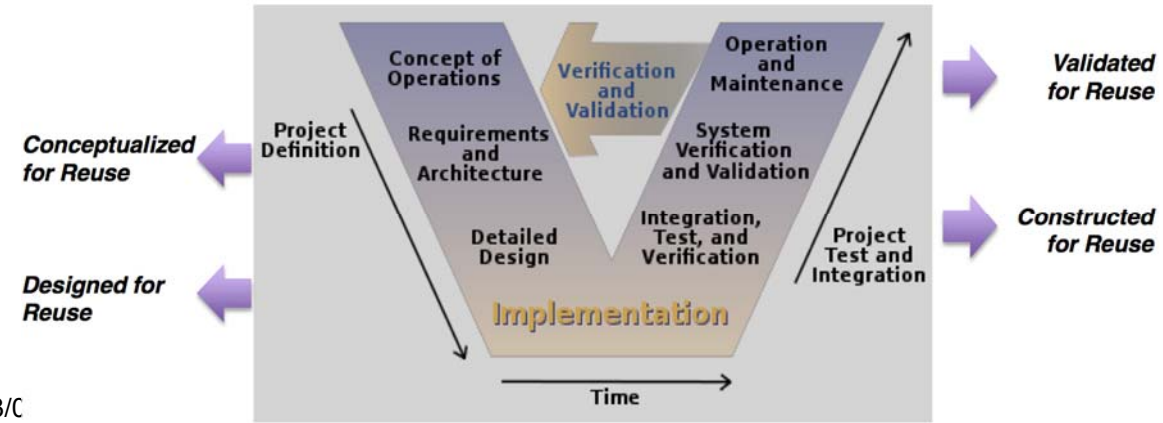
Generalized Reuse Framework*
Top Level Part 1

- **The Generalized (Systems Engineering) Reuse Framework extends the COSYSMO family of cost estimating models to account for the influence of reusing system engineering artifacts and developing them for such reuse**
- **Under this model, all system engineering effort falls under one of these types:**
  - **Development with Reuse**
  - **Development for Reuse**

***Material in this section is taken from [1].**

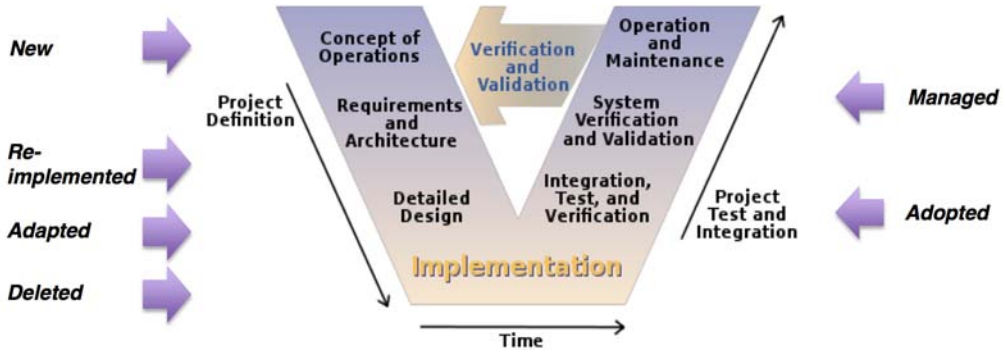
Generalized Reuse Framework: Development for Reuse

- **Development for Reuse produces artifacts intended for later reuse on projects. A completed DFR artifact may (intentionally) not be completely developed, so that it will be in one of these DFR states:**
  - **Conceptualized for Reuse (e.g., Concept of Operations document)**
  - **Designed for Reuse (e.g., component detailed design)**
  - **Constructed for Reuse (e.g., integrated component)**
  - **Validated for Reuse (e.g., validated component)**



Generalized Reuse Framework: Development with Reuse

- **Development with Reuse is project development, with reusable artifacts being brought into the product**
  - **A special case: zero reusable artifacts**
- **Each reusable artifact is included in one of these DWR states of maturity:**
  - **New (i.e., not reused)**
  - **Re-implemented (through requirements & architecture)**
  - **Adapted (through detailed design)**
  - **Adopted (through implementation)**
  - **Managed (through system verification & validation)**



Generalized Reuse Framework: Top Level Part 2

- A system engineering project to be estimated will consist of these types of effort:
  - Development with Reuse; or
  - Development for Reuse; or
  - Both, with the DFR effort typically producing some artifacts for use in the DWR effort.
- A project's estimated total system engineering effort, then, is estimated as:
  - Estimated DFR effort + estimated DWR effort
- DFR effort is estimated via an extended COSYSMO model
  - DWR effort, likewise

Generalized Reuse Framework: COSYSMO (1/2)

- COSYSMO [2] starts by computing the "size" of a system engineering project, in units of eReq ("equivalent nominal requirements")
- These artifacts are considered in the size: system requirements, system interfaces, system-critical algorithms, and operational scenarios.
- Each artifact is evaluated as being easy, nominal, or difficult.
- Each artifact is looked up in this size table to get its number of eReq, and then these are summed to get the system size:

Artifact Type	Easy	Nominal	Difficult
System Req'ts	0.5	1.0	5.0
System Interfaces	1.1	2.8	6.3
System Algs	2.2	4.1	11.5
Op Scenarios	6.2	14.4	30.0

Generalized Reuse Framework COSYSMO (2/2)

$$Size_{COSYSMO} = \sum_{artifacts} size(art\,type,\ art\,difficulty)$$

- **Size is raised to an exponent, representing diseconomy of scale, and then multiplied by factors for 14 effort multipliers and a calibration constant.**
- **This results in the following equation for a COSYSMO estimate of effort in person-months:**

$$PM_{COSYSMO} = A \cdot (Size_{COSYSMO})^{E} \cdot \prod_{j=1}^{14} EM_j$$

Generalized Reuse Framework: DFR Model Equations

- **A DFR estimate adjusts each artifact's size contribution by considering its DFR state according to this table:**

DFR State (Degree of Development)	DFR State Factor
Conceptualized for Reuse	36.98%
Designed for Reuse	58.02%
Constructed for Reuse	79.15%
Validated for Reuse	94.74%

$$Size_{DFR} = \sum_{artifacts} size(art\,type,\ art\,difficulty) \cdot DFRStateFactor(art\,state)$$

$$PM_{DFR} = A_{DFR} \cdot (Size_{DFR})^{E_{DFR}} \cdot \prod_{j=1}^{14} EM_{DFR_j}$$

Generalized Reuse Framework: DWR Model Equations

- **A DWR estimate adjusts each artifact's size contribution by considering its DWR state according to this table:**

DWR State (Maturity)	DWR State Factor
New	100.00%
Re-Implemented	66.73%
Adapted	56.27%
Adopted	38.80%
Managed	21.70%

$$Size_{DWR} = \sum_{artifacts} size(art\,type,\ art\,difficulty) \cdot DWRStateFactor(art\,state)$$

$$PM_{DWR} = A_{DWR} \cdot (Size_{DWR})^{E_{DWR}} \cdot \prod_{j=1}^{14} EM_{DWR_j}$$

COSATMO/COSYSMO Generalized Reuse Framework Topic

- **Can model be generalized/simplified by just looking at which phases of development an artifact needs to be put through? (Alstad)**
  - **I.e., just develop a per-phase cost model**
    - **Presumably separate parameters for DFR & DWR**
  - **Would need a common set of phases for DFR & DWR.**
  - **Would remove restrictions that DFR development always starts from scratch and that DWR development always goes to product completion.**

Harmonized COSYSMO 3.0 Effort Multiplier Model (1/3)

- **14 effort multipliers unchanged from COSYSMO 1.0 (Table 16 of [2]):**

Driver Name	Data Item
Requirements understanding	Subjective assessment of the system requirements
Architecture understanding	Subjective assessment of the system architecture
Level of service requirements	Subjective difficulty of satisfying the key performance parameters
Migration complexity	Influence of legacy system (if applicable)
Technology risk	Maturity, readiness, and obsolescence of technology
Documentation to match life cycle needs	Breadth and depth of required documentation
# and Diversity of installations/platforms	Sites, installations, operating environment, and diverse platforms
# of Recursive levels in the design	Number of applicable levels of the Work Breakdown Structure
Stakeholder team cohesion	Subjective assessment of all stakeholders
Personnel/team capability	Subjective assessment of the team's intellectual capability
Personnel experience/continuity	Subjective assessment of staff consistency
Process capability	CMMI level or equivalent rating
Multisite coordination	Location of stakeholders and coordination barriers
Tool support	Subjective assessment of SE tools

- **Rating levels and rating scales unchanged**

Harmonized COSYSMO 3.0 Effort Multiplier Model (2/3)

- A new, 15th effort multiplier is "System Engineering for Reuse (SEFR)"
  - I.e., is the project developing intermediate and final system engineering results to be reused on later projects?
    - Reuse for product line is one example
  - Inspired by [1]
- Assumes there is an added cost for SEFR
- Starting point for rating scale (as suggested by Boehm) is COCOMO II RUSE:
  - Low: Not for reuse
  - Nominal: Reused across project
  - High: Reused across program
  - Very High: Reused across product line
  - Extra High: Reused across multiple product lines

Summary of 2013 Meetings

- **24 September at Aerospace**
  - **Presentations on satellite cost estimation**
    - **Notably, Lisa Colabella's survey of cost data gathering for Operations & Support (see backup chart)**
- **24 October at COCOMO Forum**
  - **Started official COSATMO modeling effort**
  - **Got 1st draft of most important cost drivers, list of experts**
- **18 November at JPL**
  - **Presentations on their satellite cost models, including some operations modeling**
- **18 December at SMC**
  - **Obtained pointers to some of their operation & support data**

Summary of 2014 Meetings

- 26 February at Ground Systems Arch. Workshop
  - Obtained segments, cost drivers for ground systems
- 19 March at Annual SERC Technical Review
  - Presented status
- 9 April at BAE Systems
  - Private meeting on directions for COSYSMO 3.0
- 29 April at CSSE Annual Research Review
  - General coverage
  - Detailed discussion on directions for COSYSMO 3.0
  - Kick off COSYSMO 3.0 Working Group
- 29 July at CMU
  - SERC RT-113/RT-119 meeting
  - Overview of COSATMO

My Tentative Research Objectives

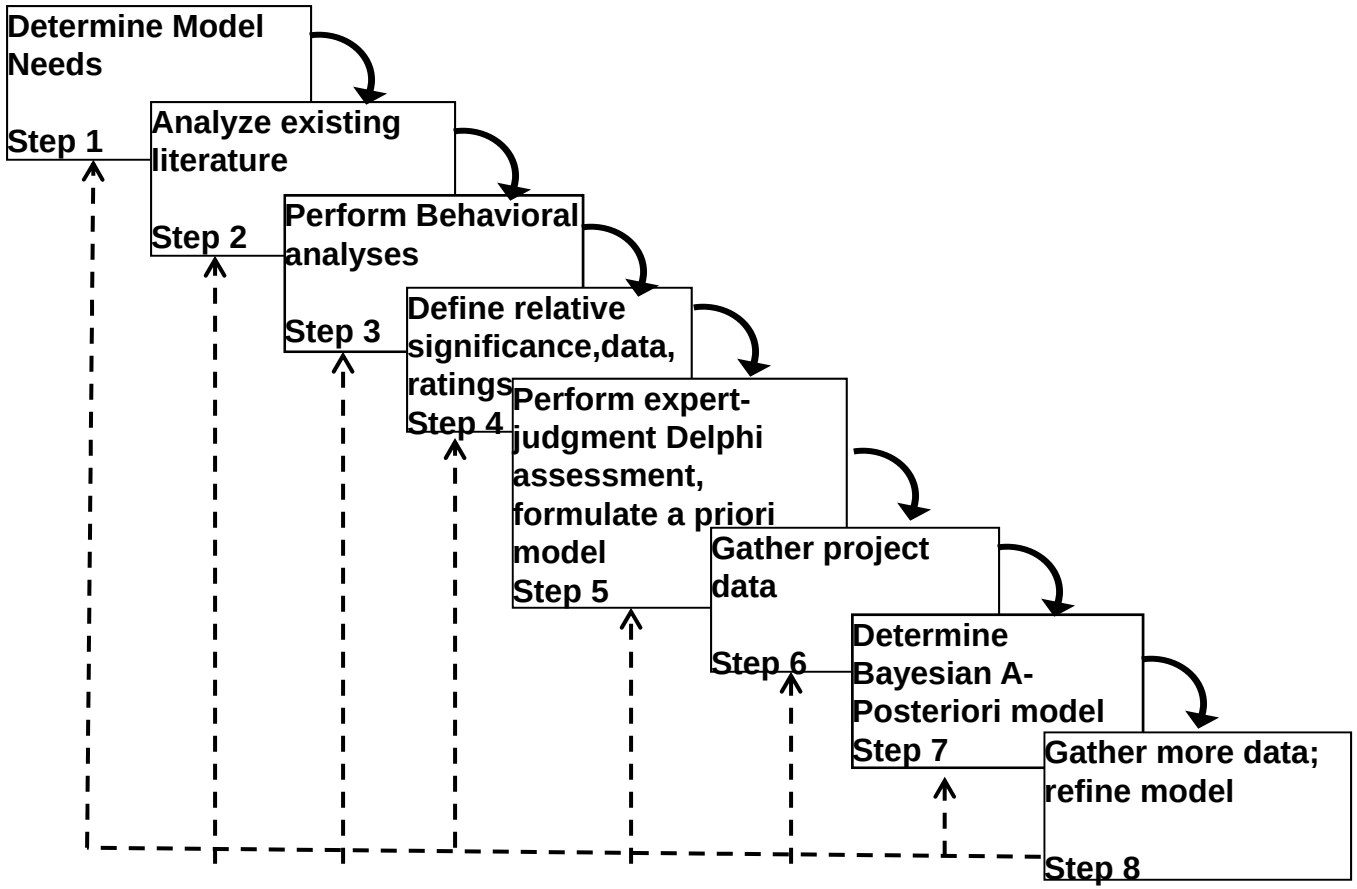
- **Provide improved cost estimation capabilities for the portions of and changing needs of space systems that are most needed and most currently tractable, including availability of calibration data. For example, SMC's main current concern is better estimation of post-deployment operations and sustainment costs.**

- **Develop a framework of cost estimation methods best suited for the various aspects of current and future space systems and other domains, such as the use of unit costing for production, acquisition, and consumables costs, and the use of activity-based costing for operations and sustainment labor costs.**

- **Prioritize the backlog of estimation models to be developed next.**

USC-CSSE Modeling Methodology

- concurrency and feedback implied

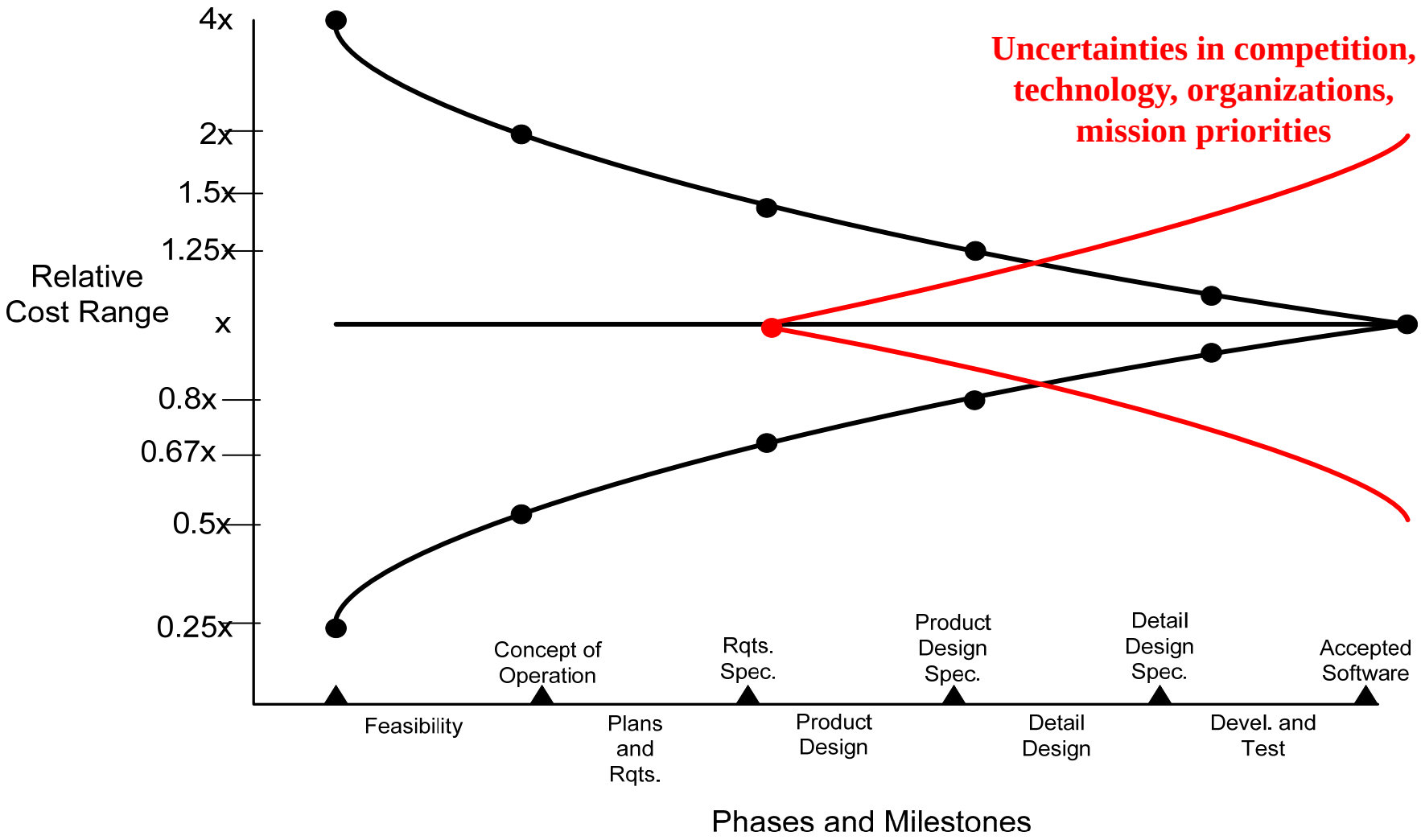


Current and Future Estimation Challenges

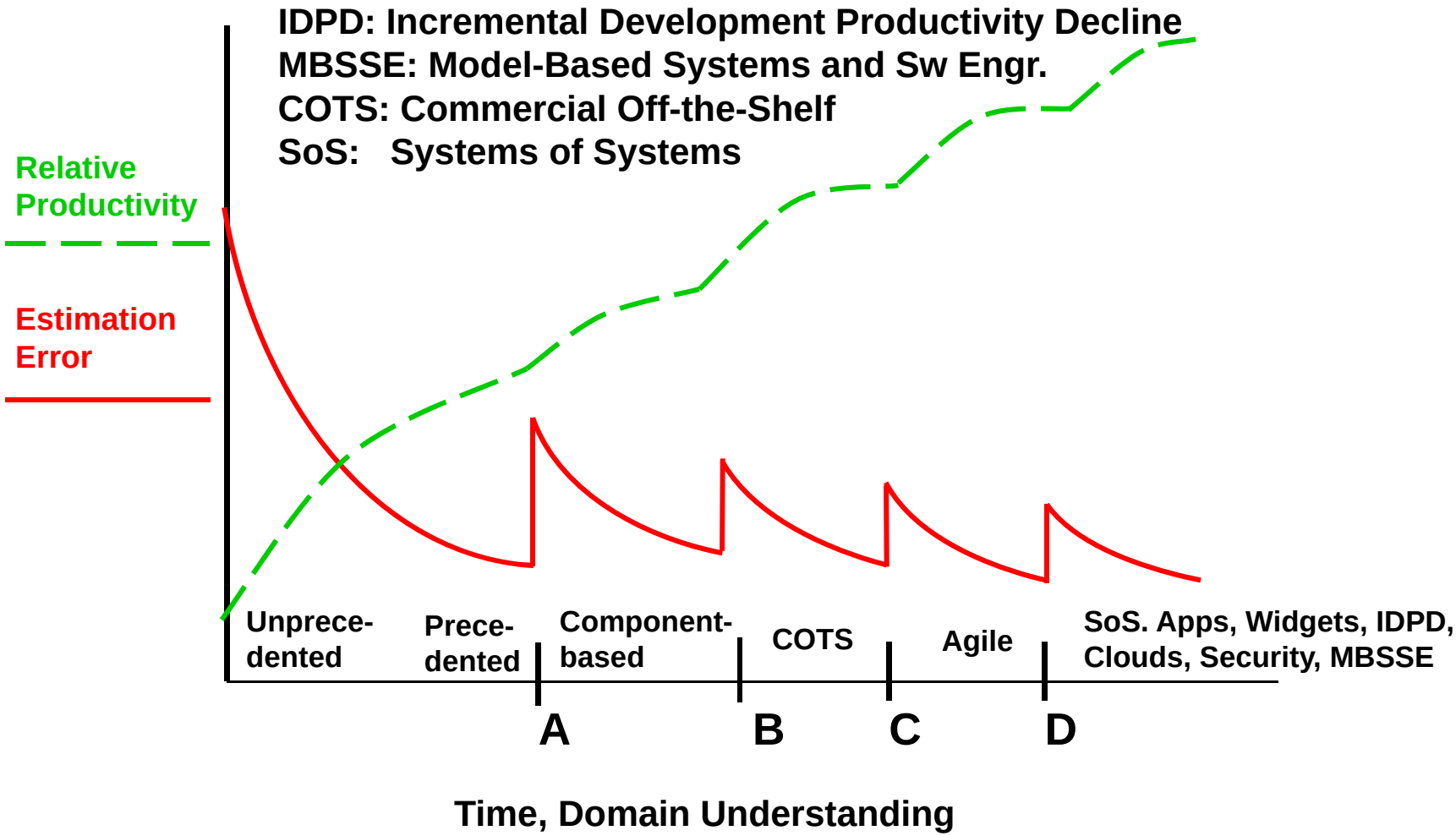
- **Emergent requirements**
  - **Cannot prespecify requirements, cost, schedule, EVMS**
  - **Need to estimate and track early concurrent engineering**
- **Rapid change**
  - **Long acquisition cycles breed obsolescence**
  - **Need better models for incremental development**
- **Net-centric systems of systems**
  - **Incomplete visibility and control of elements**
- **Model, COTS, service-based, Brownfield systems**
  - **New phenomenology, counting rules**
- **Major concerns with affordability**
  - **Multi-mission ground system challenges**

Rapid Change Creates a Late Cone of Uncertainty

– Need evolutionary/incremental vs. one-shot development



Multi-Mission Ground Systems Costing

- **Product Line Engineering**
  - Identify multi-mission commonalities and variabilities
  - Identify fully, partially sharable commonalities
  - Develop plug-compatible interfaces for variabilities
- **Product Line Costing (COPLIMO) Parameters**
  - Fractions of system fully reusable, partially reusable and cost of developing them for reuse
  - Fraction of system variabilities and cost of development
  - System lifetime and rates of change
- **Product Line Life Cycle Challenges**
  - Layered services vs. functional hierarchy
  - Modularization around sources of change
  - Version control, CTS refresh, and change prioritization
  - Balancing agilty, assurance, and affordability

Software Estimation: The Receding Horizon