Practical Software and Systems Measurement

A foundation for objective project management

COSYSMO Requirements Volatility Workshop

July 12 2010

Dr. Ricardo Valerdi
Mauricio Peña

PSM Users Group Conference
11-15 July 2011
Mystic, CT

Workshop Agenda

• *Introductions, Objectives and COSYSMO Overview*	*1:30 – 1:45*
• *Requirements Volatility Background and Preliminary Research Results*	*1:45 – 2:15 pm*
• *Delphi Survey Round #1*	*2:15 – 3:00 pm*
• *Break*	*3:00 – 3:30 pm*
• *Delphi Survey Round # 2*	*3:30 – 4:00 pm*
• *Feedback on counting rules*	*4:00 – 4:45 pm*
• *Discussion*	*4:45 – 5:00 pm*

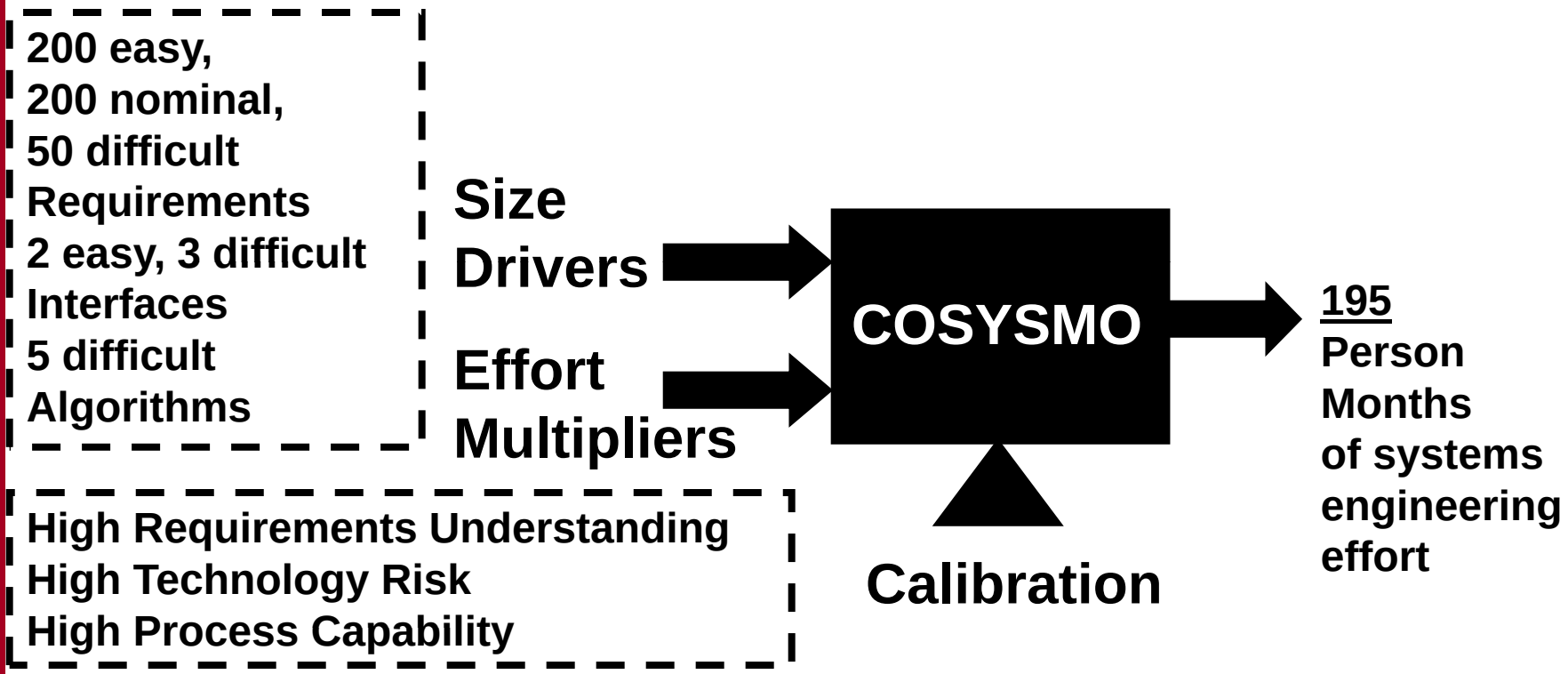
Objectives of the Workshop

- ***Learn about COSYSMO and the development of an extension of the model to account for the volatility of system requirements***
- ***Discuss the causes and effects of requirements volatility and review research results from prior workshops***
- ***Review requirements volatility base measures and measurement methods***
- ***Discuss requirements volatility data collection challenges and share lessons learned***

Intended Outputs

- Agreement on a set of requirements volatility counting rules and data collection approach
- Convergence of expert opinion on the expected level and impact of requirements volatility across the system lifecycle
- Documentation of lessons learned on requirements volatility data collection

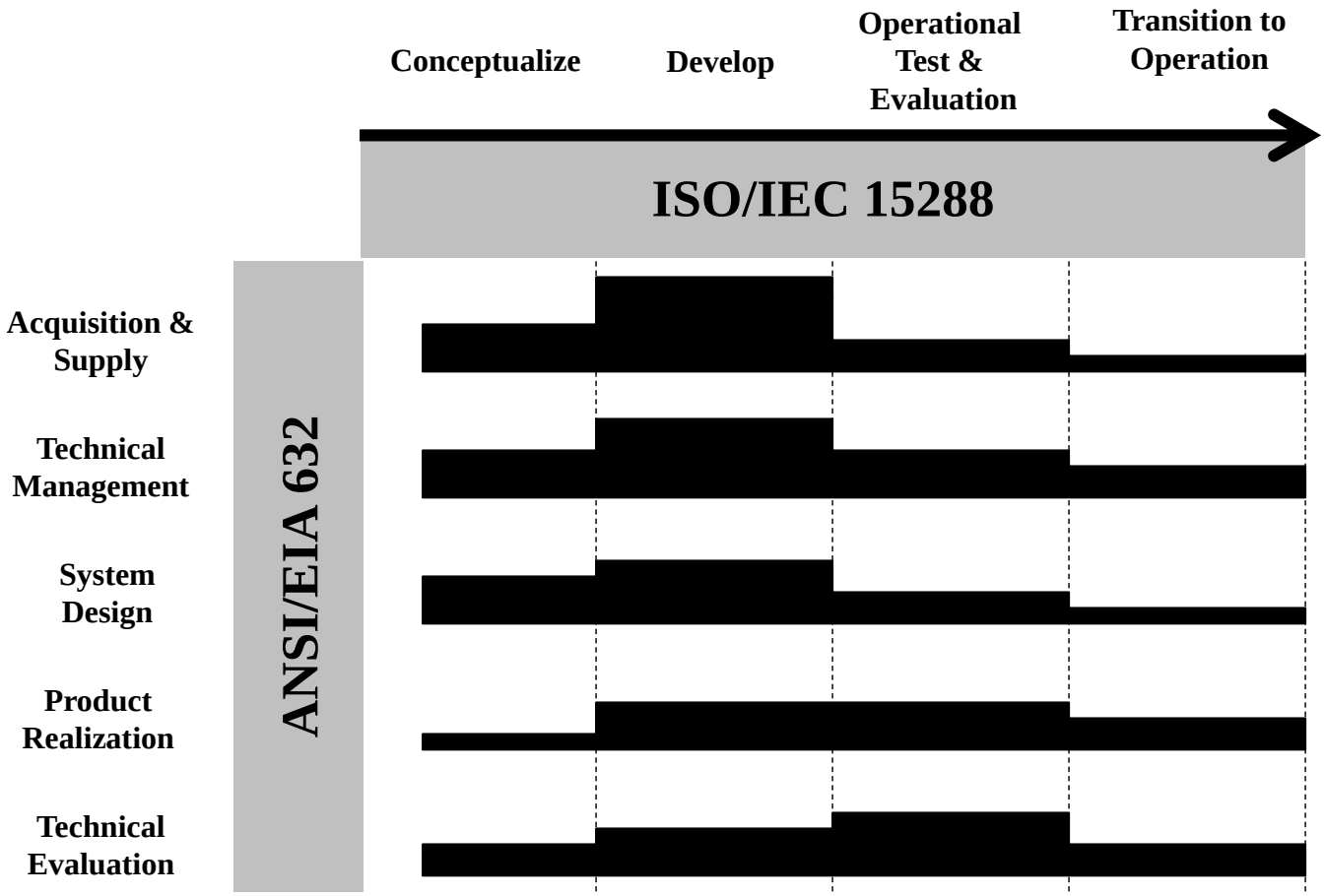
Bottom Line Up Front



Cost Driver Rating Scales

	Very Low	Low	Nominal	High	Very High	Extra High	EMR
Requirements Understanding	1.87	1.37	1.00	0.77	0.60		3.12
Architecture Understanding	1.64	1.28	1.00	0.81	0.65		2.52
Level of Service Requirements	0.62	0.79	1.00	1.36	1.85		2.98
Migration Complexity			1.00	1.25	1.55	1.93	1.93
Technology Risk	0.67	0.82	1.00	1.32	1.75		2.61
Documentation	0.78	0.88	1.00	1.13	1.28		1.64
# and diversity of installations/platforms			1.00	1.23	1.52	1.87	1.87
# of recursive levels in the design	0.76	0.87	1.00	1.21	1.47		1.93
Stakeholder team cohesion	1.50	1.22	1.00	0.81	0.65		2.31
Personnel/team capability	1.50	1.22	1.00	0.81	0.65		2.31
Personnel experience/continuity	1.48	1.22	1.00	0.82	0.67		2.21
Process capability	1.47	1.21	1.00	0.88	0.77	0.68	2.16
Multisite coordination	1.39	1.18	1.00	0.90	0.80	0.72	1.93
Tool support	1.39	1.18	1.00	0.85	0.72		1.93

Systems Engineering Effort Profile



ENTER SIZE PARAMETERS FOR SYSTEM OF INTEREST

	Easy	*Nominal*	*Difficult*
# of System Requirements			
# of System Interfaces			
# of Algorithms			
# of Operational Scenarios			

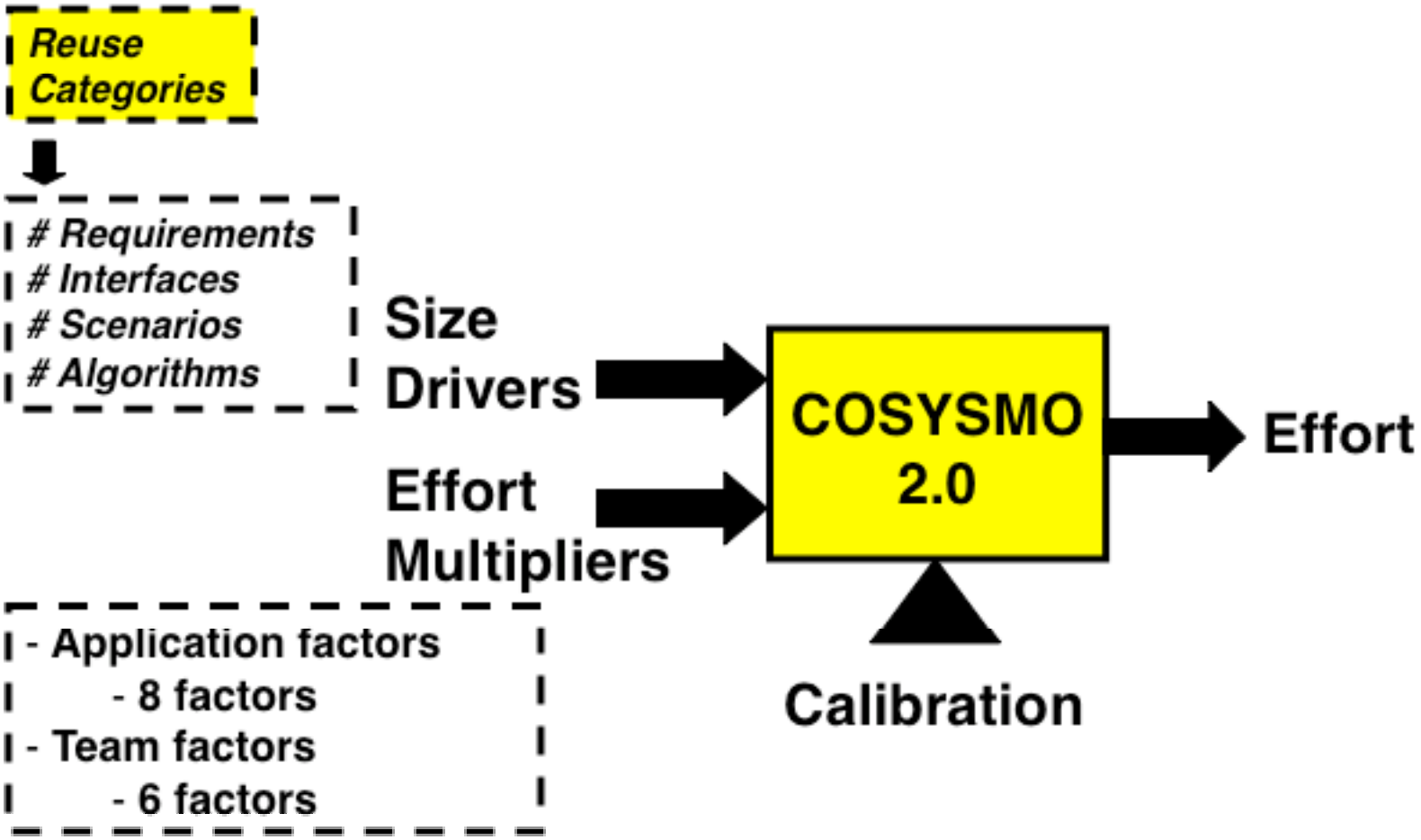
0
0
0
0
0 equivalent size

SELECT COST PARAMETERS FOR SYSTEM OF INTEREST

Requirements Understanding	**N**	1.00
Architecture Understanding	**N**	1.00
Level of Service Requirements	**N**	1.00
Migration Complexity	**N**	1.00
Technology Risk	**N**	1.00
Documentation	**N**	1.00
# and diversity of installations/platforms	**N**	1.00
# of recursive levels in the design	**N**	1.00
Stakeholder team cohesion	**N**	1.00
Personnel/team capability	**N**	1.00
Personnel experience/continuity	**N**	1.00
Process capability	**N**	1.00
Multisite coordination	**N**	1.00
Tool support	**N**	1.00
		1.00 composite effort multiplier

SYSTEMS ENGINEERING PERSON MONTHS **0.0**

COSYSMO 2.0 Operational Concept



Source: Fortune (2009)

Model Form

$$PM_{NS} = A \cdot \left[\sum_{k} \left(\sum_{r} w_r (w_{e,k}\Phi_{e,k} + w_{n,k}\Phi_{n,k} + w_{d,k}\Phi_{d,k}) \right) \right]^{E} \cdot \prod_{j=1}^{14} EM_j$$

PM_{NS} = effort in Person Months (Nominal Schedule)

A = calibration constant derived from historical project data

k = {Requirements, Interfaces, Algorithms, Scenarios}

w_x = weight for "easy", "nominal", or "difficult" size driver

r = {New, Design for Reuse, Modified, Deleted, Adopted, Managed}

w_r = weight for reuse category

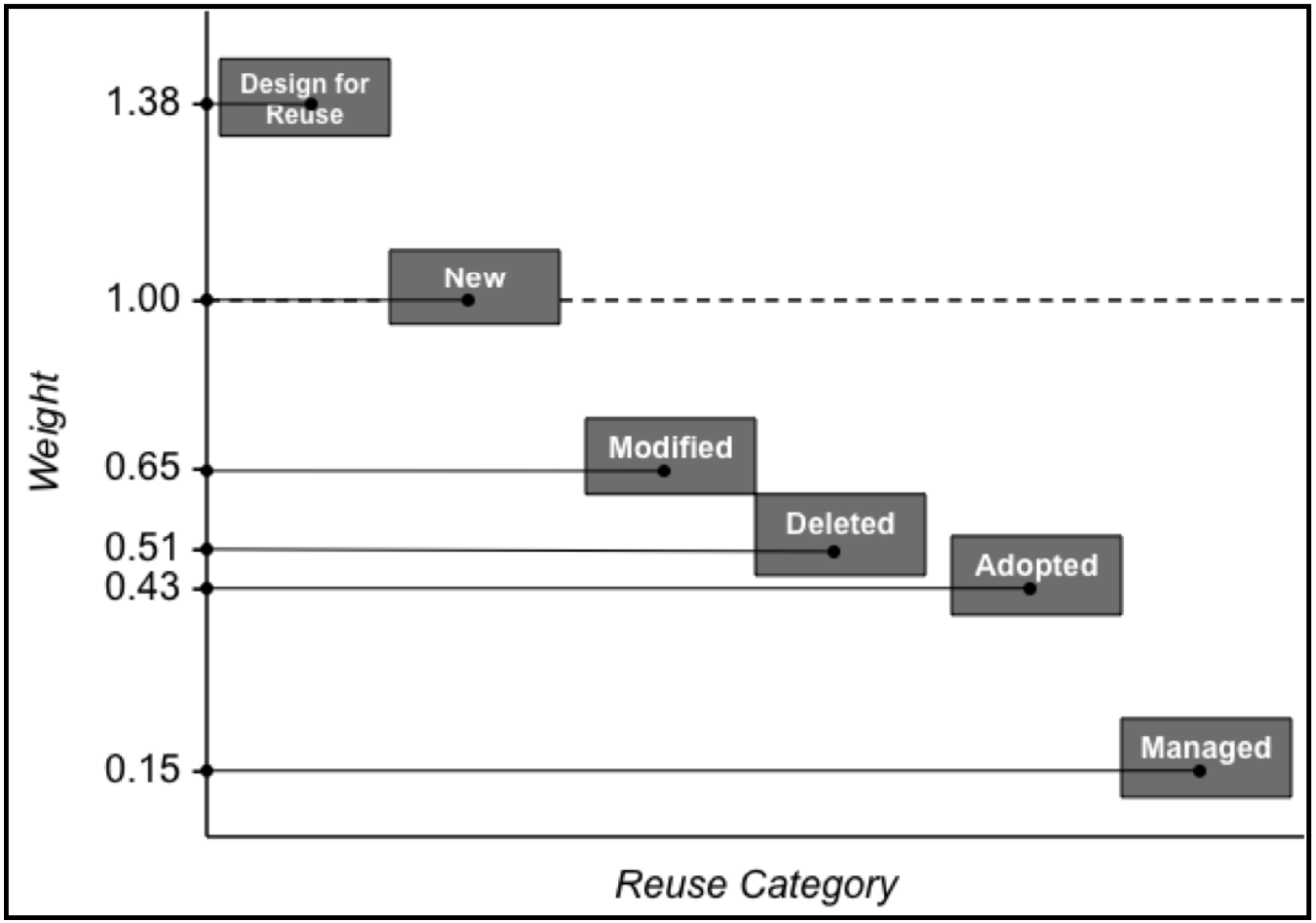
Φ_x = quantity of "k" size driver

E = represents (dis)economies of scale

EM = effort multiplier for the j^{th} cost driver.

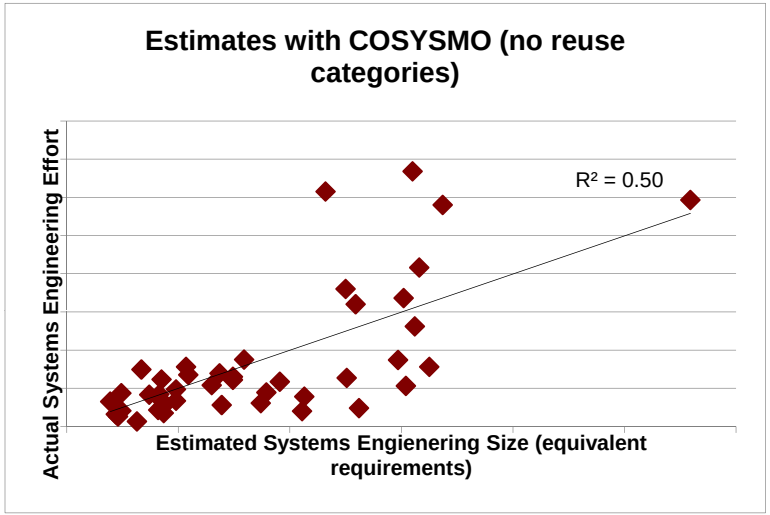
Source: Fortune (2009)

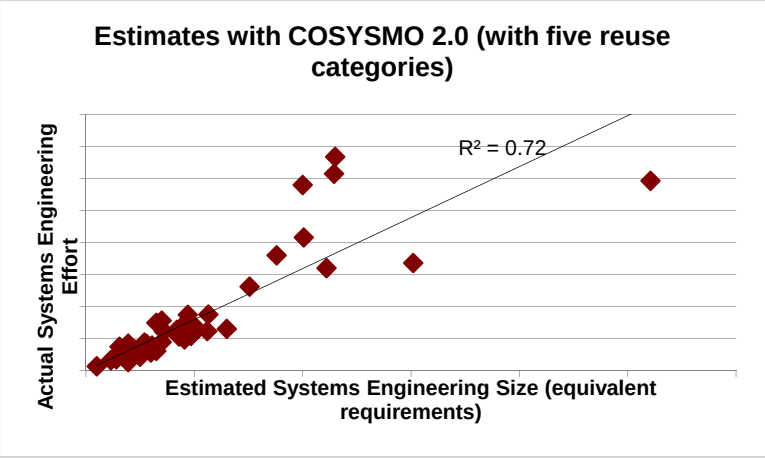
Reuse Category Weights



Source: Fortune (2009)

COSYSMO 2.0 Implementation Results





Source: Fortune (2009)

- ***Across 44 projects at 1 diversified organization***
- ***Using COSYSMO:***
  - ***PRED(.30) = 14%***
  - ***PRED(.40) = 20%***
  - ***PRED(.50) = 20%***
  - ***R^2 = 0.50***
- ***Using COSYSMO 2.0:***
  - ***PRED(.30) = 34%***
  - ***PRED(.40) = 50%***
  - ***PRED(.50) = 57%***
  - ***R^2 = 0.72***
  - ***Result: 36 of 44 (82%) estimates improved***

Importance of Understanding Requirements Volatility

- *Requirements volatility has been identified by numerous research studies as a risk factor and cost-driver of systems engineering projects[1]*
- *Requirements changes are costly, particularly in the later stages of the lifecycle process because the change may require rework of the design, verification and deployment plans[2]*
- *The Government Accountability Office (GAO) concluded in a 2004 report on the DoD's acquisition of software-intensive weapons systems that missing, vague, or changing requirements are a major cause of project failure[3]*

System developers often lack effective methods and tools to account for and manage requirements volatility

 Source: 1- Boehm (1991), 2- Kotonya and Sommerville (1995), 3- GAO-04-393

Requirements Volatility is Expected

- *Changes to requirements are a part of our increasingly complex systems & dynamic business environment*
  - *Stakeholders needs evolve rapidly*
  - *The customer may not be able to fully specify the system requirements up front*
  - *New requirements may emerge as knowledge of the system evolves*
  - *Requirements often change during the early phases of the project as a result of trades and negotiations*

Requirements volatility must be anticipated and managed

Sources: Kotonya and Sommerville (1995); Reifer (2000)

Questions that will be covered

- How do we account for the impact of requirements volatility on functional size and effort estimates?
- What counting rules should be used for requirements volatility?
  - How do we distinguish between typical changes/iterations and unplanned changes that require more effort than originally projected?
  - How do we prevent from counting the elaboration of requirements as volatility (added requirements)?
- Is it feasible to track the volatility of other size drivers (interfaces, operational scenarios, algorithms)?

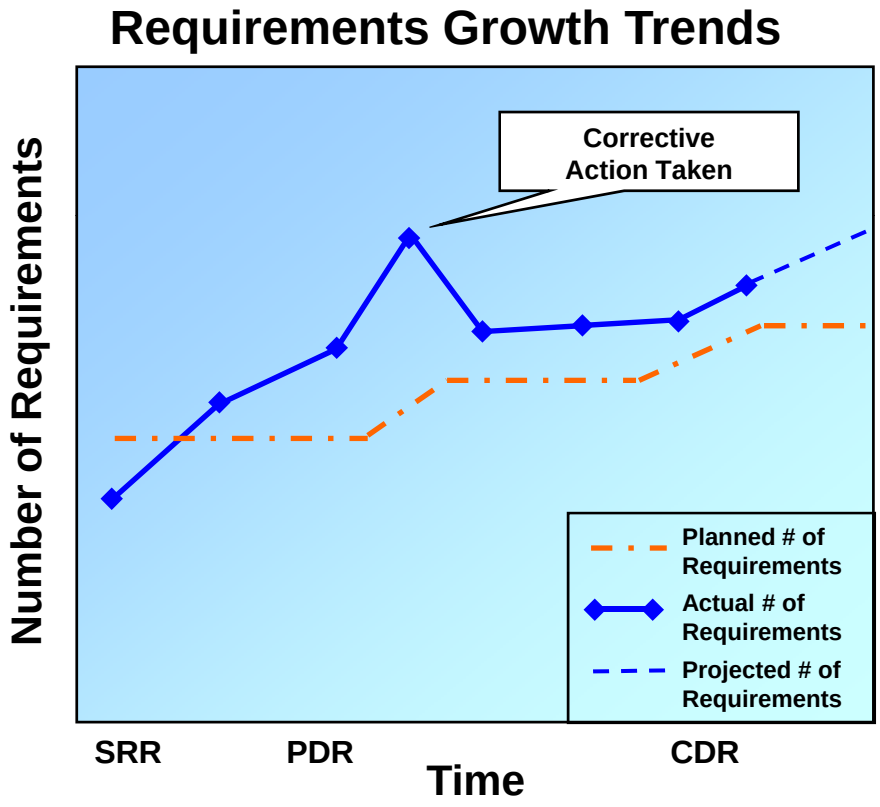
Requirements Volatility Definitions

- **Requirements volatility** ***is typically defined as the change in requirements (added, deleted, and modified) over a given time interval***
- ***Also known as:***
- **Requirements creep:** ***An increase in scope and number of system requirements***
- **Requirements churn:** ***Instability in the requirements set – requirements are modified or re-worked without necessarily resulting in an increase in the total number of requirements***

Source: MIL-STD-498

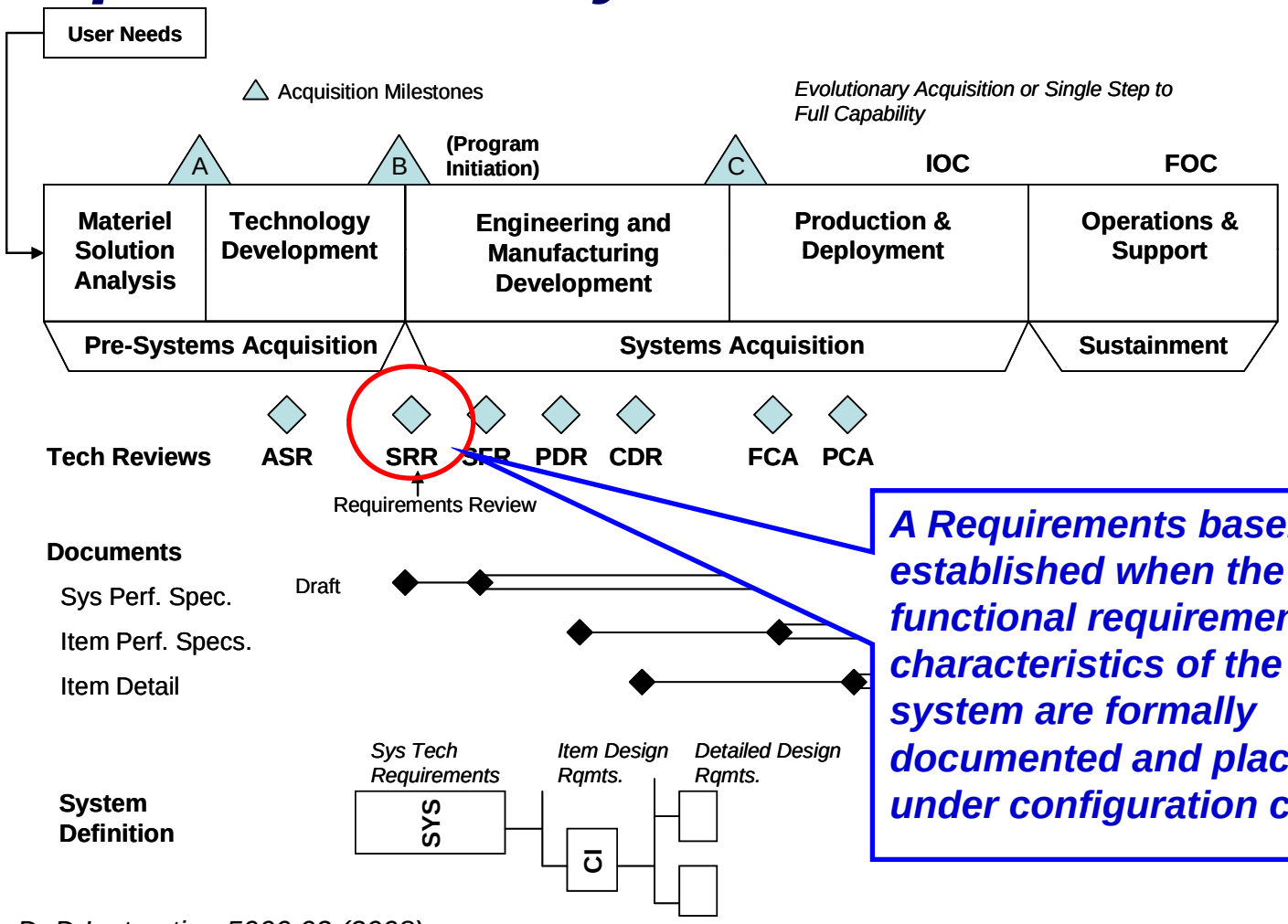
Requirements Trends as a Leading Indicator of Project Performance

- *Evaluates trends in the growth, change, completeness and correctness of the system requirements.*
- *It helps to determine the stability and completeness of the system requirements which could potentially impact project performance*



Source: Systems Engineering Leading Indicators Guide, Version 2.0, 2010

Systems Engineering Reviews and Acquisition Lifecycle Phases



A Requirements baseline is established when the functional requirements and characteristics of the system are formally documented and placed under configuration control

Source: DoD Instruction 5000.02 (2008)

Requirements Volatility Measures

- **Base Measures**
  - Number of requirements
  - Number of added, deleted or modified requirements
- **Measurement methods**
  - Count the number of requirements changes from Engineering Change Proposals (ECPs)
  - Count the number and type of changes using a requirements management tool such as the Dynamic Object Oriented Requirements System (DOORS)
  - Use a text differencing tool to determine the number and type of requirements changes between two versions of a specification
- **Measurement Frequency**
  - Typically monthly or as required

Source: Systems Engineering Leading Indicators Guide, Version 2.0, 2010

Requirements Derived Measures

% Requirements Volatility =

((# of requirements added + # of requirements deleted + # of requirements modified) / total # of requirements)

% Requirements Growth =

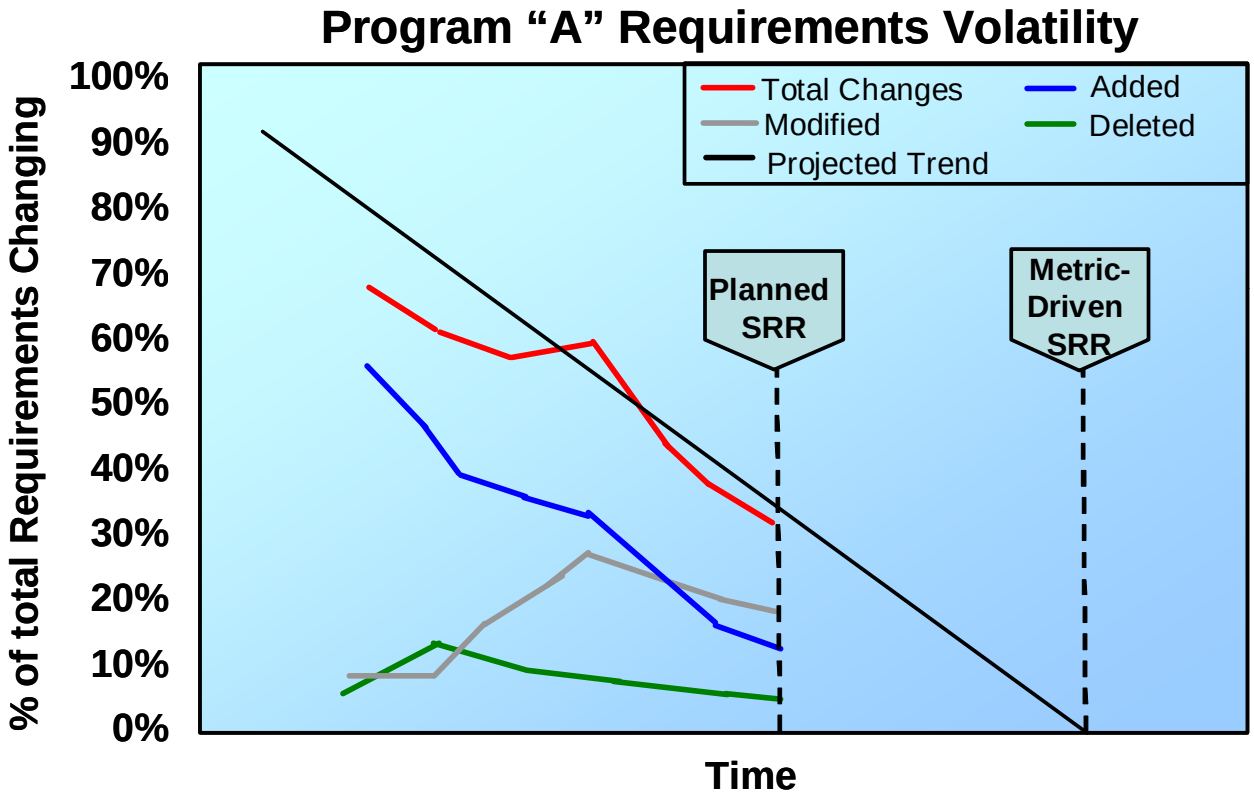
$$\frac{\text{(\# of requirements in current baseline - \# of requirements in previous baseline)}}{\text{(\# of requirements in previous baseline) X 100}}$$

% Requirements Modified =

(# of requirements modified / total # of requirements)

Source: Systems Engineering Leading Indicators Guide, Version 2.0, 2010

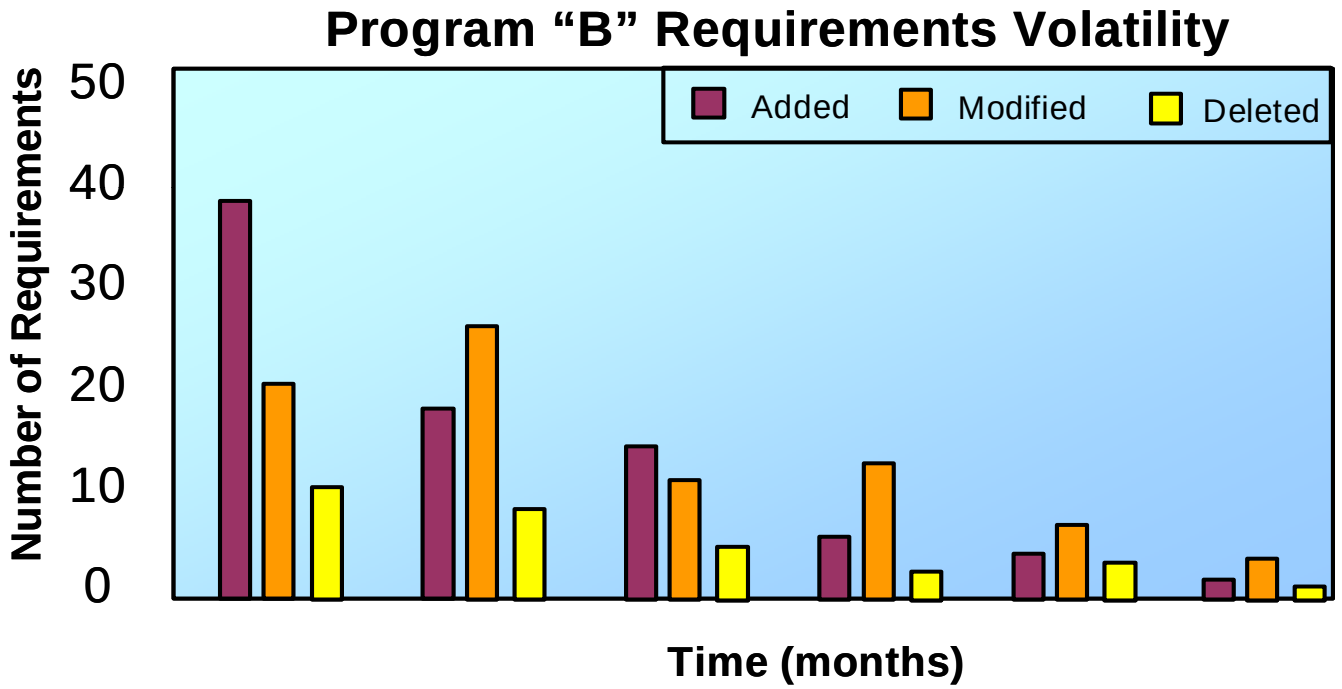
Requirements Volatility Metrics (1 of 3)



Volatility expressed as a % of the total number of requirements

Systems Engineering Leading Indicators Guide, Version 2.0, 2010

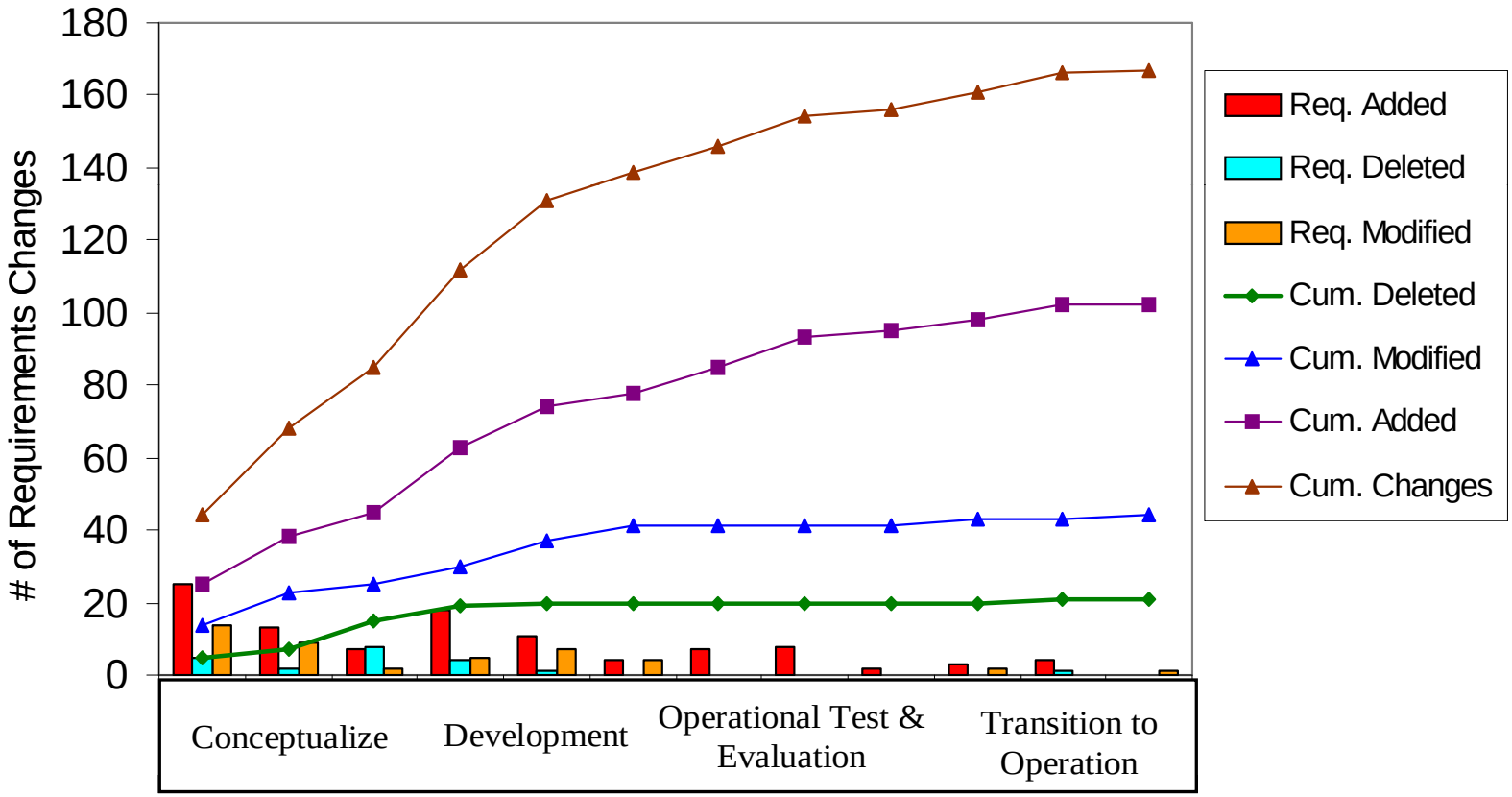
Requirements Volatility Metrics (2 of 3)



Volatility metrics track the number and type of changes over time

Source: Hammer, T., Huffman, L., and Rosenberg, L. (1998).

Requirements Volatility Metrics (3 of 3)

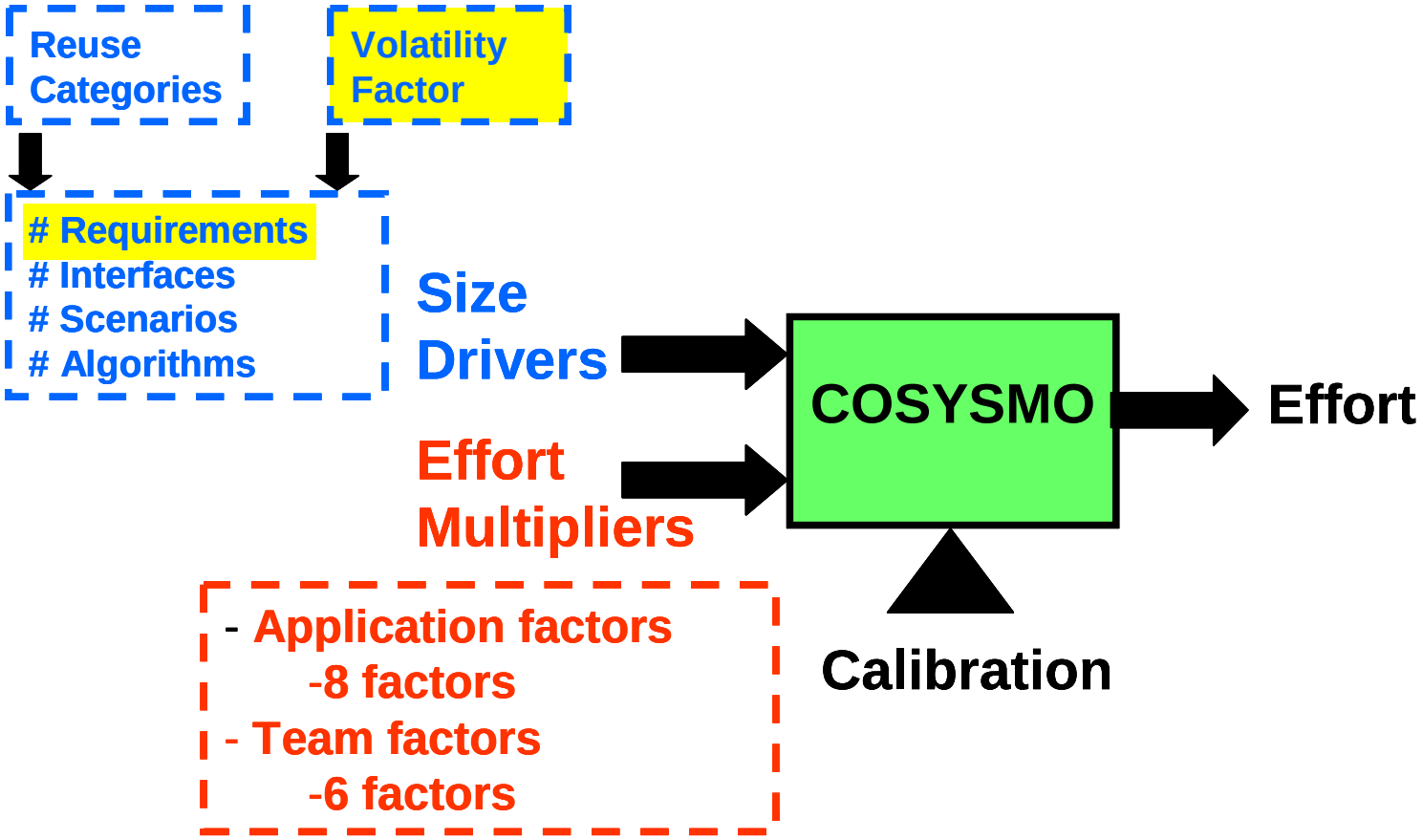


Implications to COSYSMO

- During the development of COSYSMO, volatility was identified as a relevant adjustment factor to the model's size drivers
- However, there was insufficient data to incorporate volatility effects into the initial version of the model
- The primary objective of the research is to complete the requirements volatility extension to COSYSMO within the existing structure and scope of the model

Source: Valerdi (2005)

COSYSMO Volatility Factor



Source: Fortune (2009)

Are volatility effects already adequately captured in COSYSMO?

- Requirements volatility may have a significant impact on the functional size of the project
  - This is significant because the size of the project represents the majority of the model's explanatory power
  - Accounting for volatility in the size driver will have a more direct impact on the effort estimate
- Some potential sources of volatility (i.e. requirements understanding) are captured in the COSYSMO effort multipliers, but the direct effect on size is not captured
- Other sources of volatility are not addressed at all

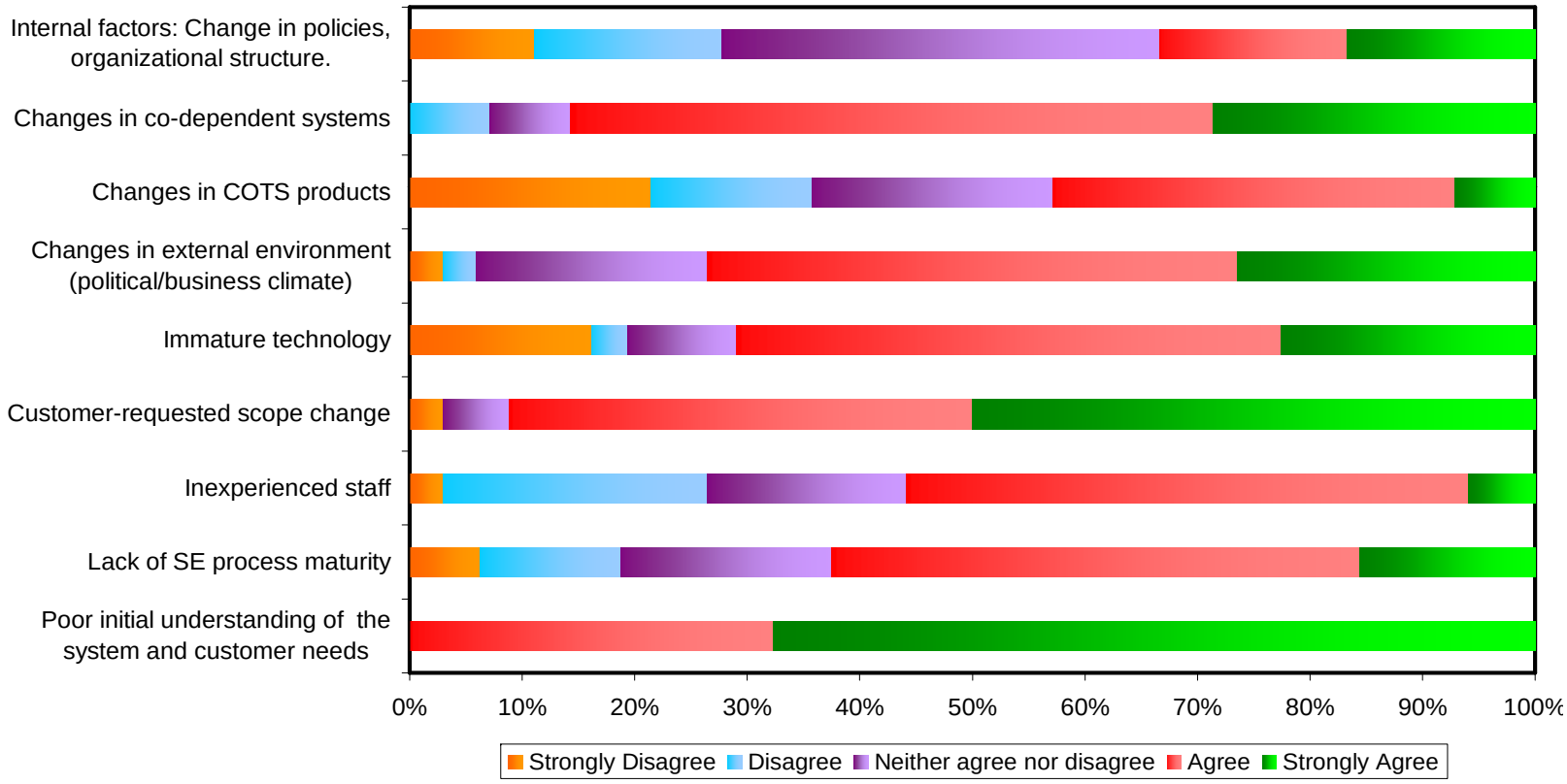
Observations from the Literature

1. Requirements volatility is caused by an identifiable set of project and organizational factors
2. The level of requirements volatility is a function of the system life cycle phase
3. Requirements volatility is correlated with an increase in project size and cost
4. The cost / effort impact of a requirements change increases the later the change occurs in the system life cycle
5. The impact of requirements volatility varies depending on the type of change: added, deleted, or modified

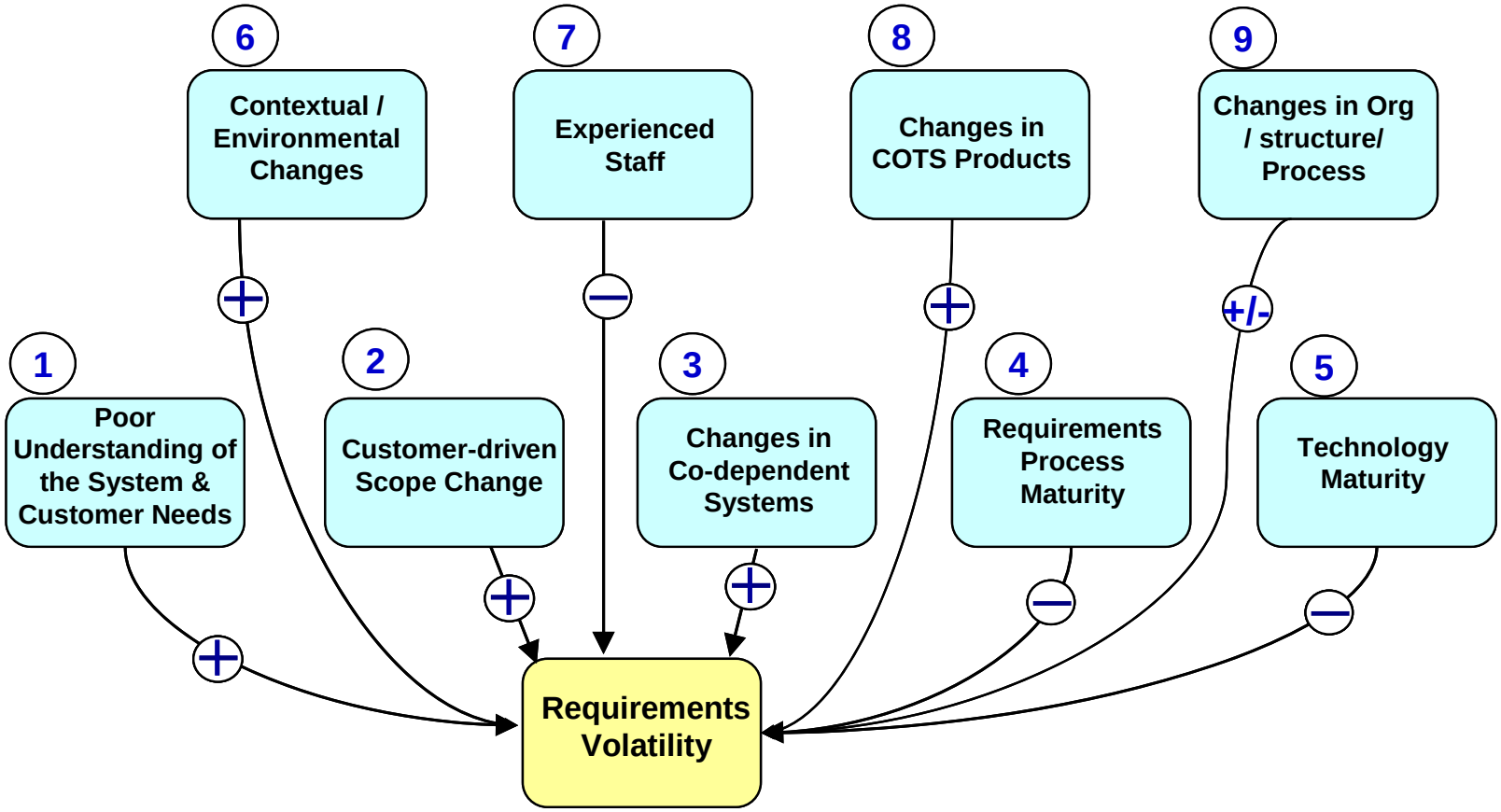
Overview of Research to Date

- *Data were collected through surveys and interviews of subject-matter experts in four different workshops*
- *A variety of industries were represented with an emphasis on aerospace & defense*
- *Exploratory survey administered at:*
  - *Workshop # 1: 2010 USC-CSSE Annual Research Review*
  - *Workshop # 2: 2010 LAI Knowledge Exchange Event*
  - *Workshop # 3: 2010 Practical Software and Systems Measurement (PSM) Users Group Conference*
  - *Workshop # 4: University of Southern California 25th Annual COCOMO Forum*
  - *Workshop # 5: 2011 USC-CSSE Annual Research Review*

Potential Causes of Requirements Volatility (N = 38)

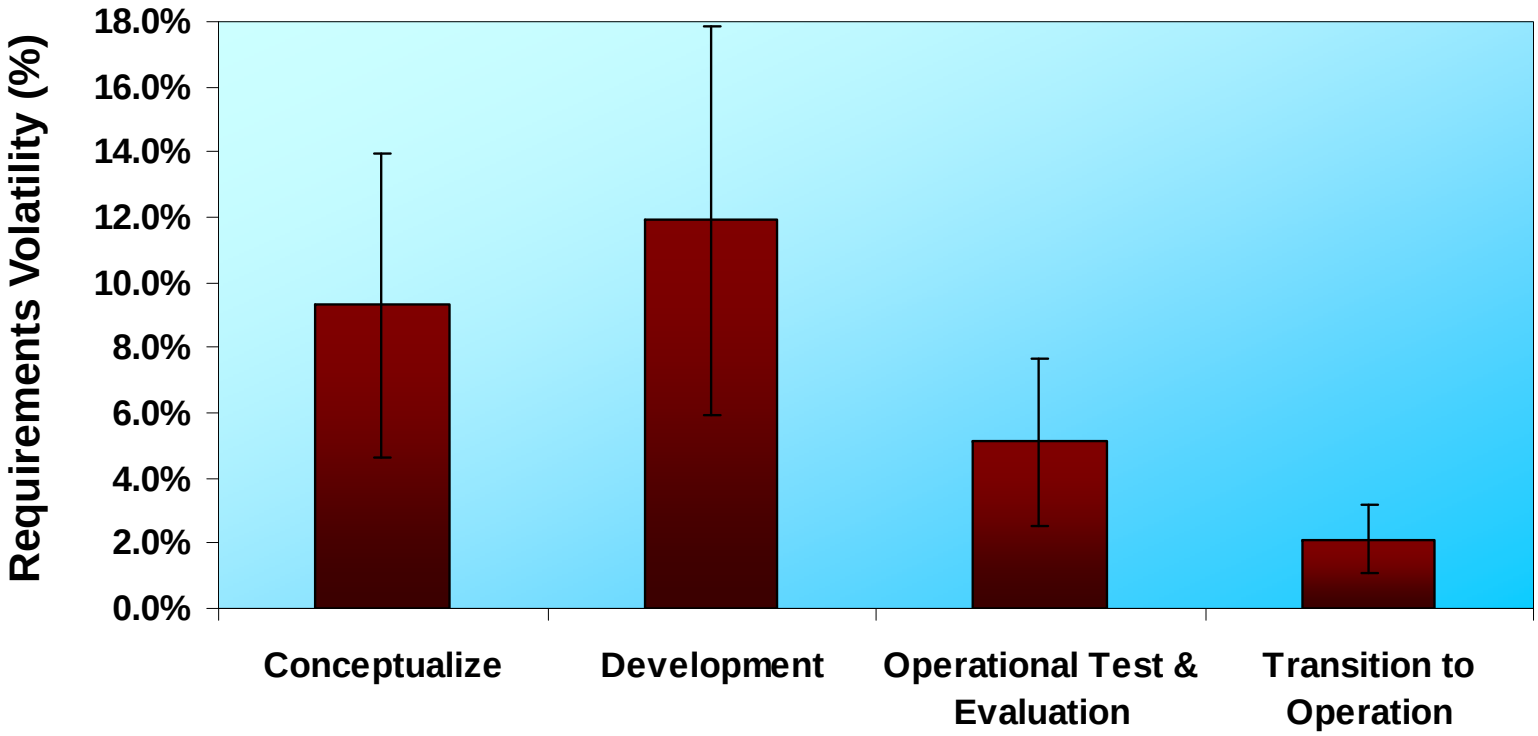


Requirements Volatility Causal Model Diagram

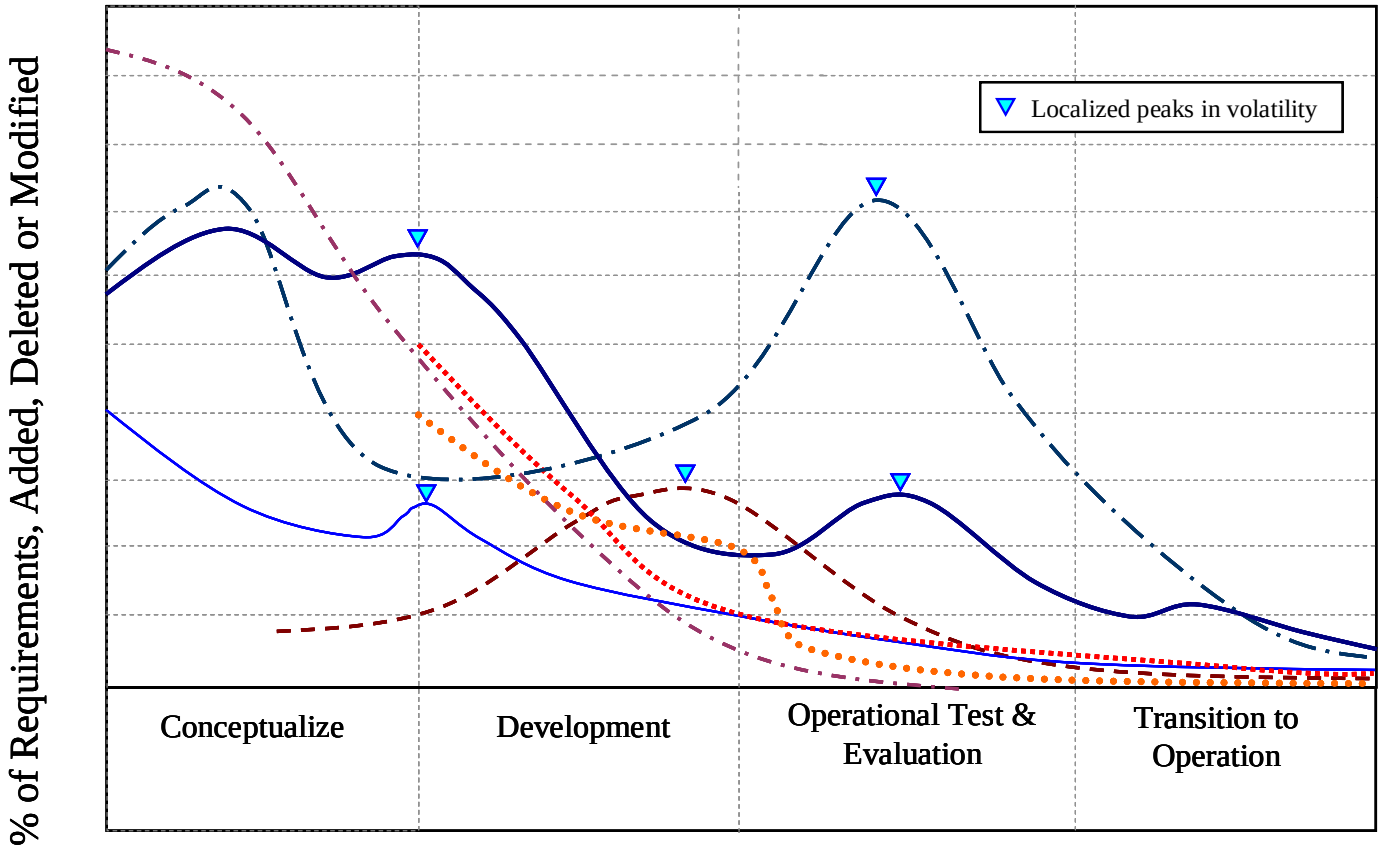


Sources: Kotonya and Sommerville (1995); Hammer et al. (1998); Malaiya and Denton (1999); Stark et al. (1999); Houston (2000); Zowghi and Nurmuliani (2002); Kulk and Verhoef 2008; Ferreira, S., Collofello, J., Shunk, D., and Mackulak, G. (2009)

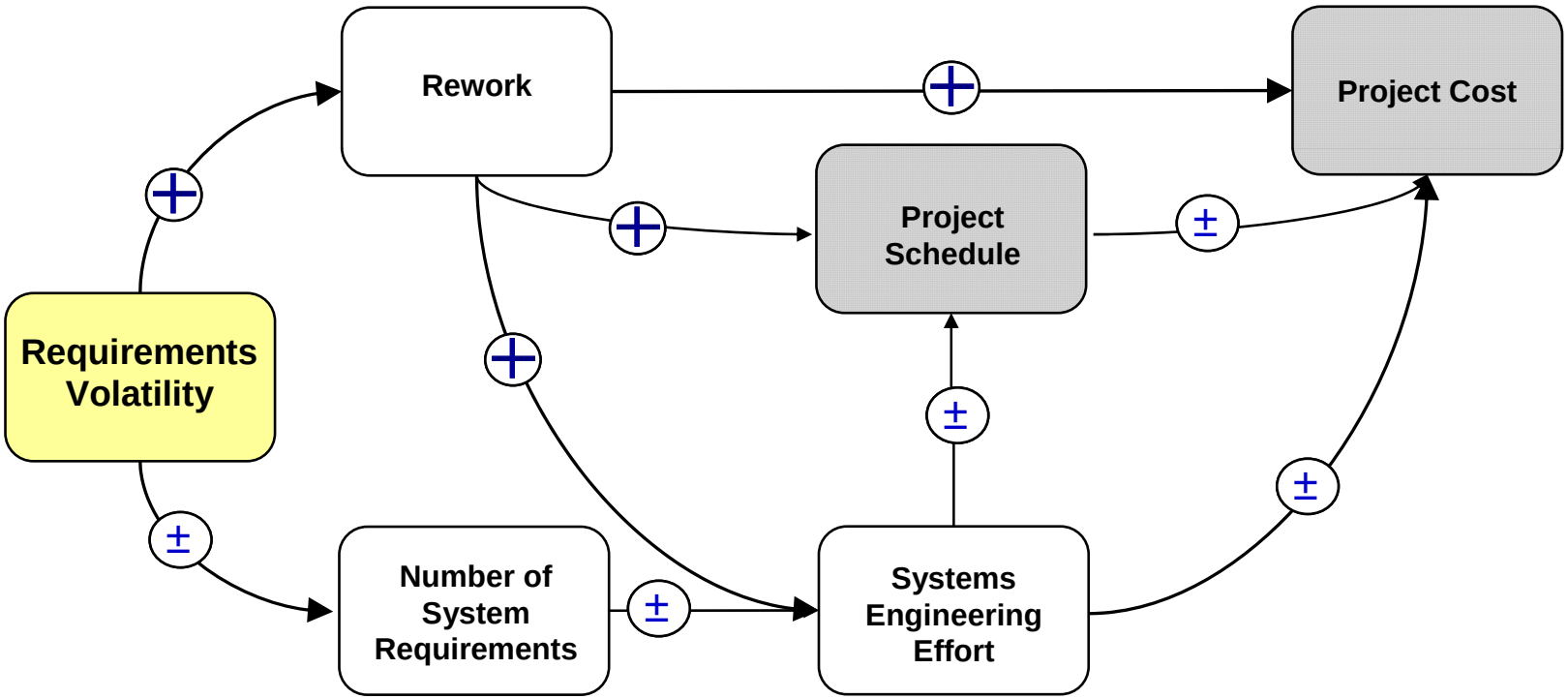
Expected Level of Requirements Volatility per Lifecycle Phase (N = 27)



Requirements Volatility Life Cycle Profile (N = 9)

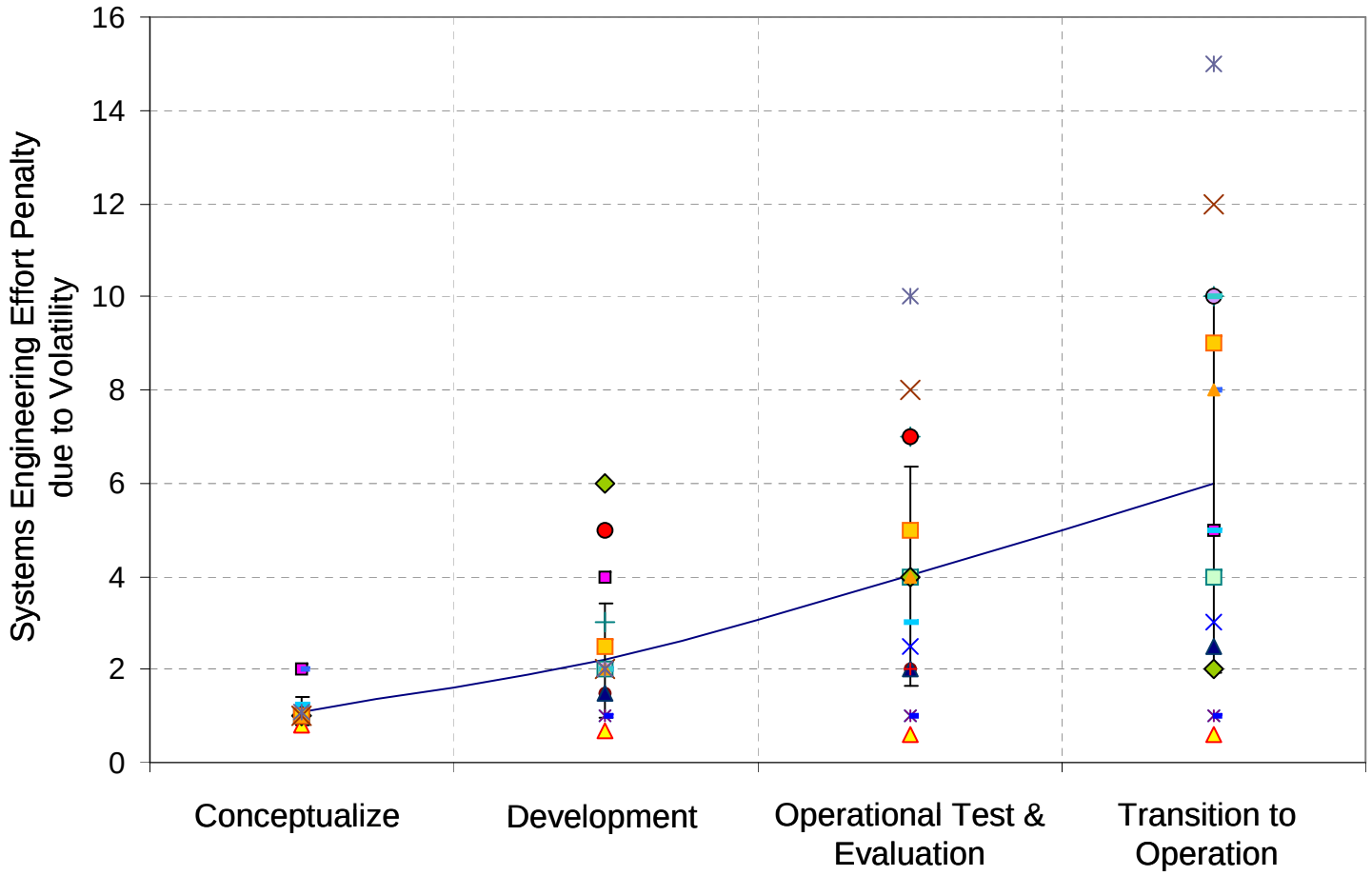


Impacts of Volatility Causal Model Diagram

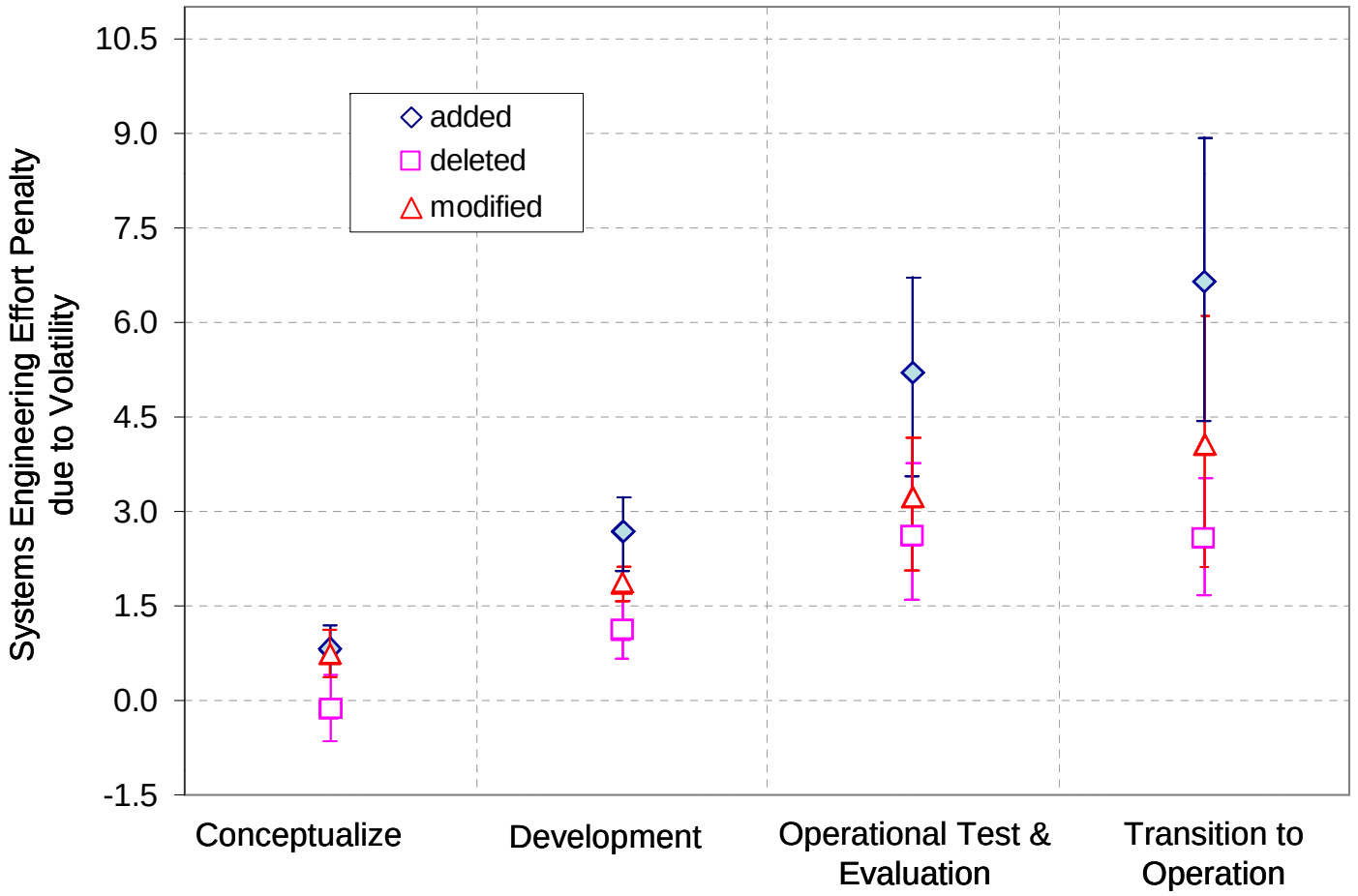


Sources: Kotonya and Sommerville (1995); Hammer et al. (1998); Malaiya and Denton (1999); Stark et al. (1999); Houston (2000); Zowghi and Nurmuliani (2002); Kulk and Verhoef 2008; Ferreira, S., Collofello, J., Shunk, D., and Mackulak, G. (2009)

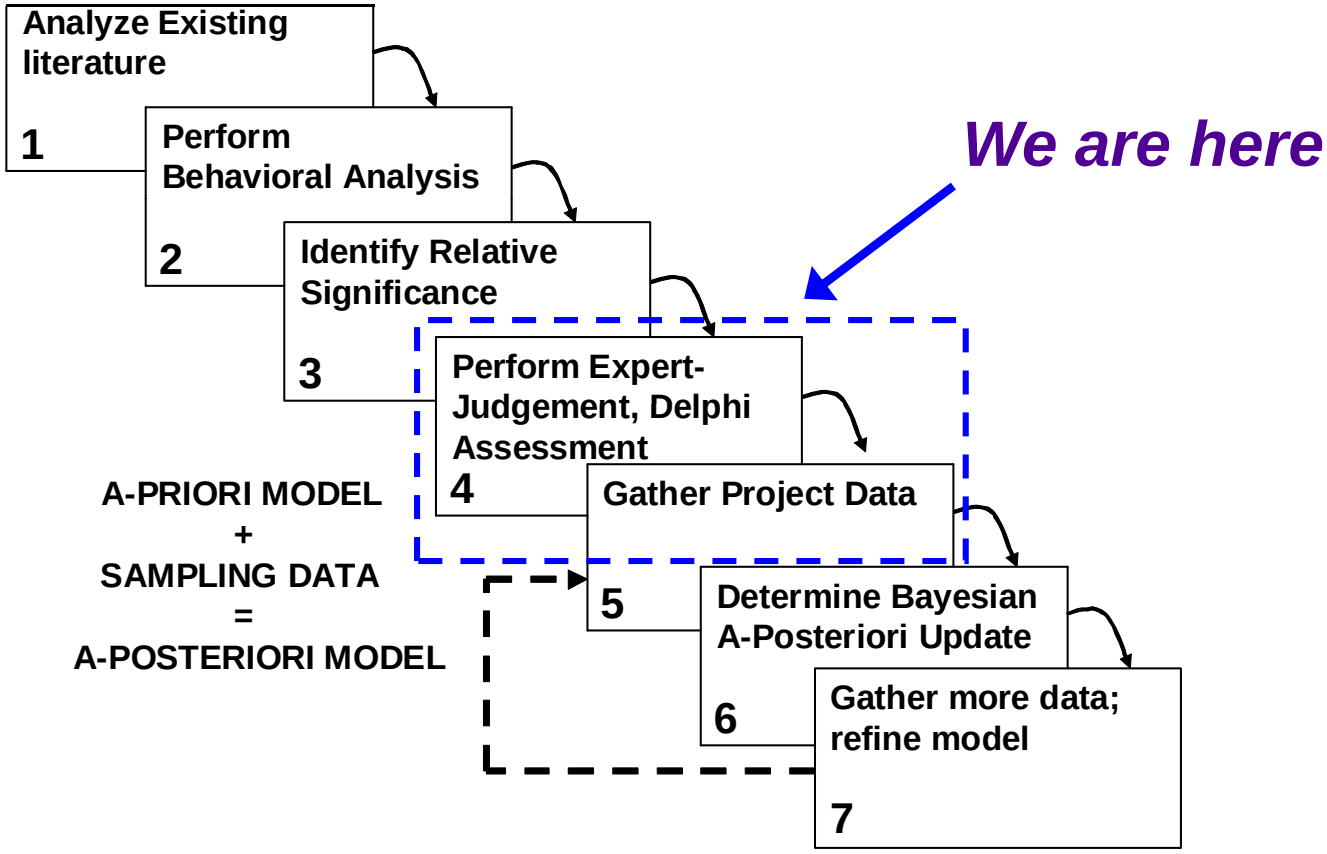
Life cycle Effort Penalty due to Volatility (N = 27)



Effort Penalty per Change Category (N = 27)



Research Methodology



Boehm, B. (1981). Software Engineering Economics. Prentice Hall.

Wideband Delphi Exercise

1. ***Please fill out the survey forms to make an assessment on:***
   - ***Level of Requirements Volatility Across life cycle phases***
   - ***Cost penalty factor due to added, deleted, or modified requirements across lifecycle phases***
2. ***The results will be collected and tabulated for analysis (Round # 1)***
3. ***The inputs, remaining anonymous, will be displayed with descriptive statistics***
4. ***The group will discuss the results and their variability***
5. ***After the discussion, we will ask you to fill out the forms again and steps 2 through 4 will be repeated***

Requirements Volatility Measures

Requirements Size Driver Definition

This driver represents the number of requirements for the system-of-interest at a specific level of design. The quantity of requirements includes those related to the effort involved in system 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 may have effort associated with is such as V&V, functional decomposition, functional allocation, etc.

System requirements can typically be quantified by counting the number of applicable shalls/wills/shoulds/mays in the system or marketing specification

Source: Valerdi (2005)

Requirements Counting Rules

- Determine the system of interest
- Decompose system objectives, capabilities, or measures of effectiveness into requirements that can be tested, verified, or designed
  - Different systems will exhibit different levels of requirements decomposition depending on the application domain, customer's ability to write good system requirements, and the functional size of the system
- Provide a graphical or narrative representation of the system of interest and how it relates to the rest of the system
- Count the number of requirements in the system/marketing specification or the verification test matrix for the level of design in which systems engineering is taking place in the desired system of interest
-

Source: Valerdi (2005)

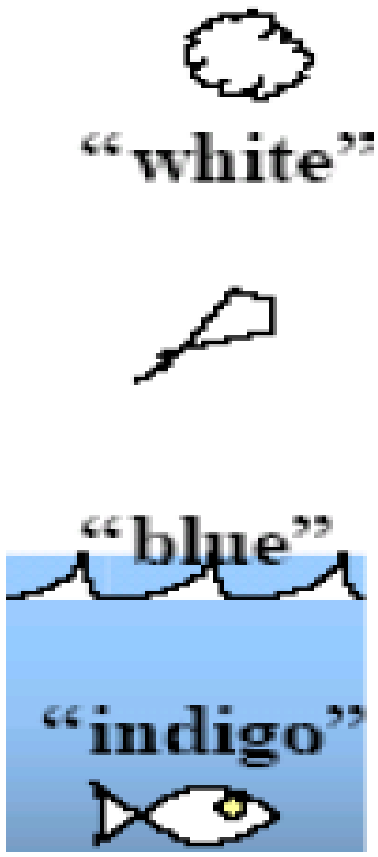
Requirements Volatility Measures

- ***Base Measures***
  - ***Number of requirements changes***
  - ***Number of added, deleted or modified requirements***
- ***Measurement methods***
  1. ***Count the number of requirements changes from Engineering Change Proposals (ECPs)***
  2. ***Count the number and type of changes using a requirements management tool such as the Dynamic Object Oriented Requirements System (DOORS)***
  3. ***Use a text differencing tool to determine the number and type of requirements changes between two versions of a specification***

Requirements Volatility Counting Rules

- We need to formulate specific counting rules for the volatility of the requirements size driver
- Questions for discussion:
  - How do we distinguish between typical changes/iterations and unplanned changes that require more effort than originally projected?
  - How do we prevent from counting the elaboration of requirements as volatility (added requirements)?
  - How do we avoid counting administrative changes (typos, etc.)?
  - How do we identify major changes in scope that could disrupt the results?
  - If using DOORS, do we count changes to all objects?

Requirements Decomposition Framework



- *Sky level*
  - *Top-level objective*
  - *Summary use case*
- *Kite level*
  - *Additional information as to how the sky-level objective is to be will be satisfied*
- *Seal level*
  - *User level task*
  - *Environment in which the developer interacts with the stakeholder*
- *Underwater level*
  - *Detailed design and implementation*

Sources: Cockburn (2001); Valerdi (2005)

Requirements Decomposition Example

- *Objective: Broadcast television signals over the Continental United States (CONUS) compatible with 18 inch receive antennas*

Too High?

1. *The system shall be able to receive a Ku-band signal from the customer ground station and downlink the signal to CONUS coverage with a minimum Equivalent isotropically radiated power (EIRP) of 30 dBm*

Just Right

1.1 The system shall have a pointing accuracy of 0.5 degrees

1.2 The system shall radiate a minimum of 3000 W of RF power

Too Low?

1.2.1 The system shall be able to produce 4000 W of DC power

1.2.2 The system shall be able to store 500 W-Hr of energy

1.2.3 The system shall be able to dissipate 2000 W of heat

1.2.3.1 The radiator surface area shall be greater than…

1.2.3.2 The radiator surface properties shall be…….

1.2.3.3 Metal surfaces shall be blanketed…….

Proposed Guidelines

- Count only requirements changes after the requirement set is baselined and under configuration control (post SRR, PDR)
- Count only changes to requirements for the system-of-interest at a specific level of design (use COSYSMO requirements counting rules)
- Identify disruptive changes in scope / re-baselining in the data collection instrument
- Use counting logic rules to exclude multiple changes to a requirement in a given reporting period (avoid counting editorial changes)

Questions on the Approach

- Should we attempt to count changes per type of requirement (easy, nominal, difficult)?
  - Is it feasible?
  - Does it add value to the analysis?
- Do you keep metrics on the volatility of the other size drivers?
  - Operational scenarios
  - System interfaces
  - Algorithms
- Would measuring the impact of requirements volatility envelope the impact of changes to the other size drives?

References

- Boehm, B. (1991). Software Risk Management: Principles and Practices. IEEE Software 8(1), pp 32-41.
- Ferreira, S., Collofello, J., Shunk, D., and Mackulak, G. (2009). "Understanding the effects of requirements volatility in software engineering by using analytical modeling and software process simulation." The Journal of Systems and Software. Vol. 82, pp 1568-1577.
- Fortune, J. (2009). Estimating systems engineering reuse with the constructive systems engineering cost model (COSYSMO 2.0). Doctoral Dissertation. University of Southern California, Industrial and Systems Engineering Department.
- GAO-04-393 (2004). Report to the Committee on Armed
- Houston, Dan X. (2000). A Software Project Simulation Model for Risk Management, Ph.D. Dissertation, Arizona State University
- INCOSE Systems Engineering Handbook, Version 3, INCOSE, June 2006
- ISO/IEC (2008). ISO/IEC 15288:2008 (E) Systems Engineering - System Life Cycle Processes.
- Kotonya, G., Sommerville, I., (1998). Requirements Engineering: Processes and Techniques. John Wiley and Sons, Ltd.
- MIL-STD-498. 1994. Software Development and Documentation. U.S. Department of Defense.
- Roedler, G. and Rhodes, D. (2007). Systems engineering leading indicators guide. Version 1. Massachusetts Institute of Technology, INCOSE, and PSM.
- Valerdi, R. (2005). The constructive systems engineering cost model (COSYSMO). Doctoral Dissertation. University of Southern California, Industrial and Systems Engineering Department.
- Zowghi, D. and Nurmuliani, N. (2002). A Study of the Impact of Requirements Volatility on Software Project Performance. Proceedings of the Ninth Asia-Pacific Software Engineering Conference

Call for Participation

- *In order to complete the requirements volatility extension of COSYSMO, we are seeking industry data for engineering projects in terms of:*
  - *Systems engineering effort actuals (labor hours)*
  - *Requirements volatility: the number of requirements, added, deleted, and modified added after the requirements baseline*
- *By providing these data your organization will benefit by:*
  - *Improving its ability to estimate the impact of requirements changes on project cost*
  - *Calibrating and tailoring the updated Model for your application domain*
- *USC-CSSE and LAI at MIT have proven processes in place to ensure the confidentiality and protection of the data with its Corporate Affiliates and Consortium Members*
- *Contact:*
  - *Mauricio E. Pena [mauricip@usc.edu]*
  - *Ricardo Valerdi [rvalerdi@mit.edu]*