# Practical Software and Systems Measurement

## A foundation for objective project management

## COSYSMO Requirements Volatility Workshop

July 27 2010

Dr. Ricardo Valerdi
Mauricio Peña

PSM Users Group Conference
26-30 July 2010
New Orleans, LA





# Workshop Agenda

| | |
|---|---|
| • Introductions & Objectives | 1:30 – 1:45 pm |
| • COSYSMO Overview & Reuse Research Results | 1:45 – 2:00 pm |
| • SE Leading Indicators & Requirements Volatility Background | 2:00 – 2:15 pm |
| • Causal Model and Feedback | 2:15 – 2:30 pm |
| • Survey Results | 2:45 – 3:00 pm |
| • Break | 3:00 – 3:30 pm |
| • Implications to COSYSMO | 3:30 – 3:45 pm |
| • Survey Exercise | 3:45 – 4:30 pm |
| • Outbrief and Discussion | 4:45 – 5:00 pm |

## Objectives of the Workshop

- Learn about COSYSMO and the latest research results in systems engineering reuse
- Provide a forum to discuss requirements volatility thresholds and metrics
- Present an overview of the causes of requirements volatility and its impact on systems engineering effort
- Obtain feedback on a proposed extension to COSYSMO to incorporate a requirements volatility cost factor
- Provide an opportunity for participants to exchange lessons learned on requirements volatility and influence the direction of future research

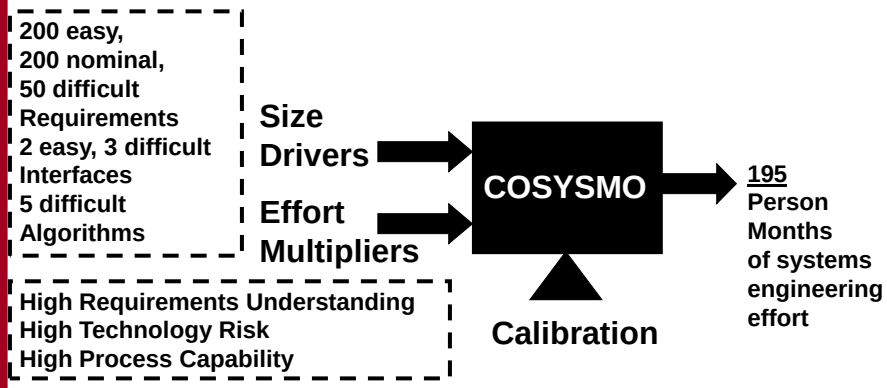




## Intended Outputs

- Feedback on a causal model that relates technical, organizational and contextual project factors to requirements volatility
- Profile of the expected level of requirements volatility as a function of system type and lifecycle phase
- Validation of the "ease of change" curve over the system lifecycle
- Feedback on the COSYSMO requirements volatility extension

# Bottom Line Up Front



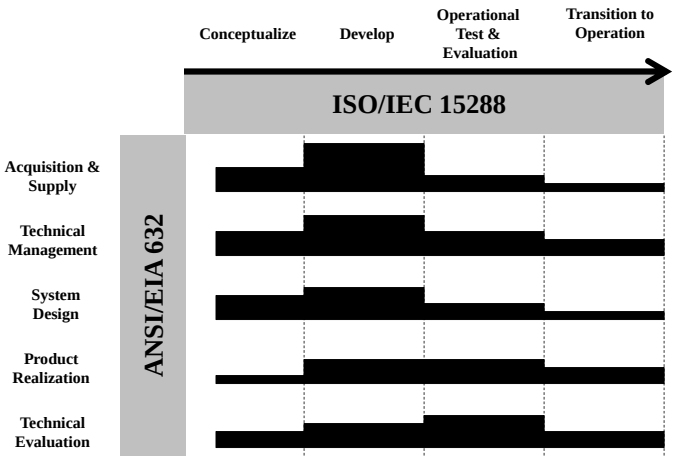





# Cost Driver Rating Scales

| | 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








COSYSMO 1.0

CONSTRUCTIVE SYSTEMS ENGINEERING COST MODEL

© Ricardo Valerdi, University of Southern California

ENTER SIZE PARAMETERS FOR SYSTEM OF INTEREST

| | Easy | Nominal | Difficult | |
|---|---|---|---|---|
| # of System Requirements | | | | 0 |
| # of System Interfaces | | | | 0 |
| # of Algorithms | | | | 0 |
| # of Operational Scenarios | | | | 0 |
| | | | | 0 |

equivalent size

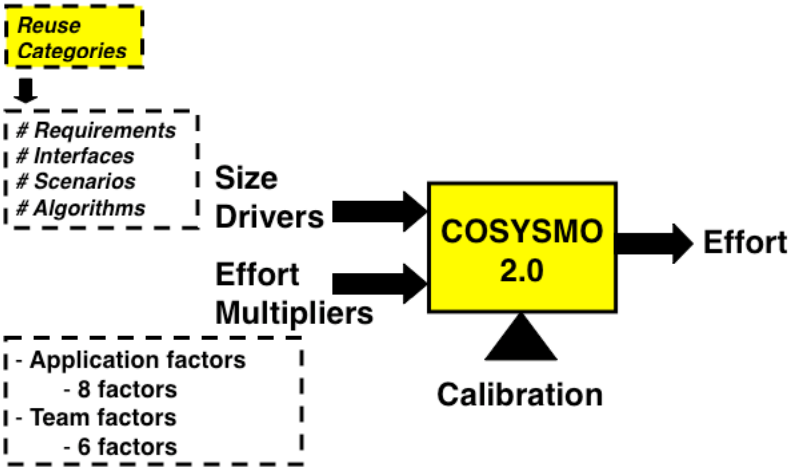
SELECT COST PARAMETERS FOR SYSTEM OF INTEREST

| | | |
|---|---|---|
| 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




Based on 2009 dissertation by Dr. Jared Fortune





# 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}

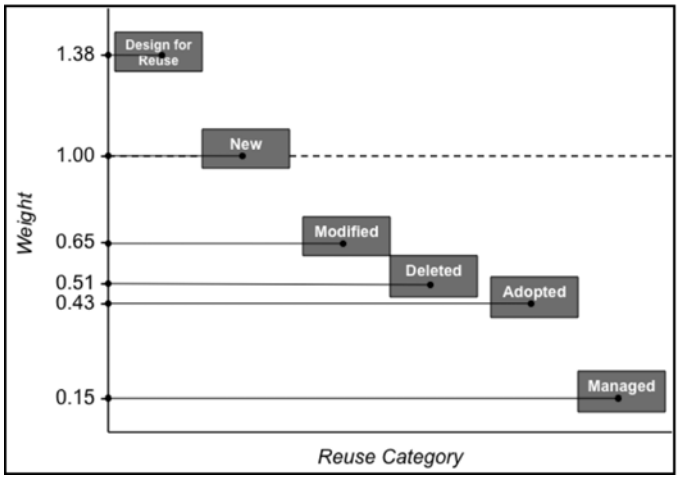
$w_r$ = weight for reuse category

$\Phi_x$ = quantity of "k" size driver

E = represents (dis)economies of scale

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

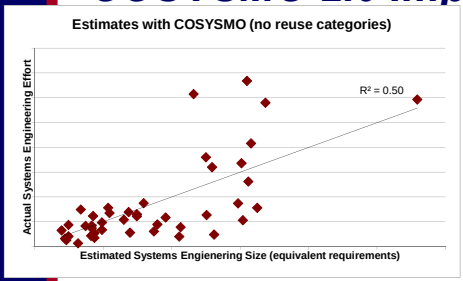
# Reuse Category Weights

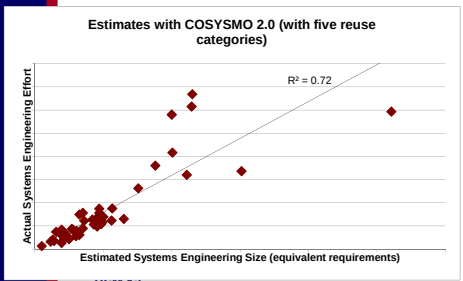






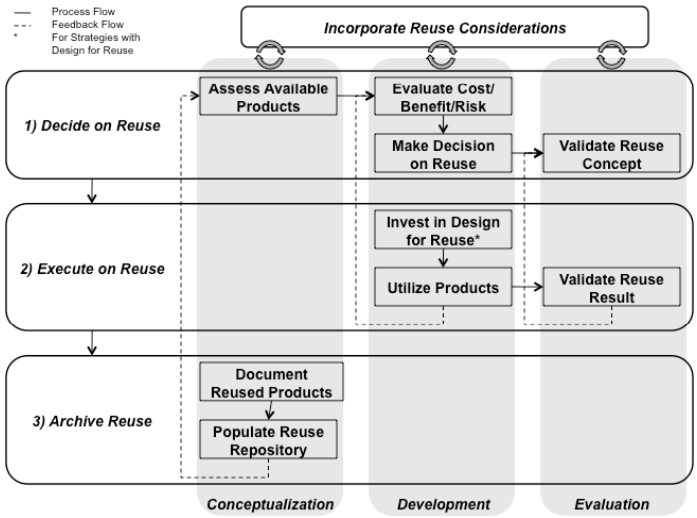

# COSYSMO 2.0 Implementation Results




- *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*

# Reuse Framework








# SE Leading Indicators Guide

**Developed and Published by Members of**

***Leading Indicators are defined as "measures for evaluating the effectiveness of the systems engineering activities on a program in a manner that provides information about impacts that are likely to affect the system or program performance objectives."***

Rhodes, D., Valerdi, R., and Roedler, G. (2009). "Systems engineering leading indicators for assessing program and technical effectiveness." *Systems Engineering* Vol. 12 (No. 1), pp 21-35.

# SE Leading Indicators

1. Requirements Trends
2. System Definition Change Backlog
3. Interface Trends
4. Requirements Validation Trends
5. Requirements Verification Trends
6. Work Product Approval Trends
7. Review Action Item Closure Trends
8. Risk Exposure Trends
9. Risk Treatment Trends
10. Technology Maturity Trends
11. Technology Measurement Trends
12. SE Staffing and Skills Trends
13. Process Compliance Trends
14. Facility and Equipment Availability Trends
15. Defect/Error Trends
16. System Affordability Trends
17. Architecture Trends
18. Schedule and Cost Pressure

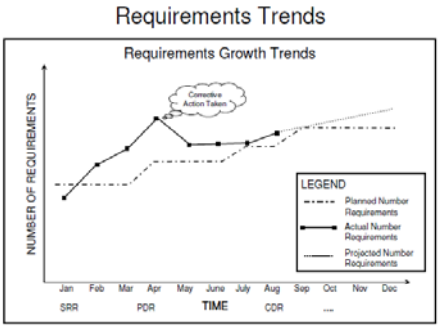
*Systems Engineering Leading Indicators Guide, Version 2.0, 2010*





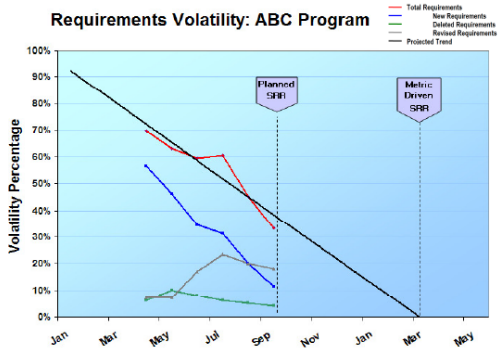
# Requirements Trends Leading Indicator

- 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




*Systems Engineering Leading Indicators Guide, Version 2.0, 2010*

# Requirements Volatility as a Leading Indicator




- *This graph depicts the rate of change of requirements over time as compared to the projected trend and can be used to predict readiness for the Systems Requirements Review (SRR)*

*Systems Engineering Leading Indicators Guide, Version 2.0, 2010*





# Requirements Volatility Definitions

***Requirements volatility*** *is the change in requirements (added, deleted, and modified) over a given time interval*

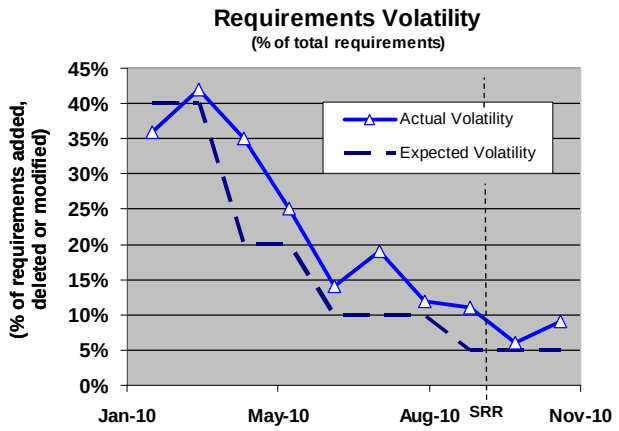
*Also known as:*

***Requirements creep:*** *An increase in scope and/or number of system requirements*

***Requirements churn:*** *Instability in the requirements set – requirements are frequently modified or reworked without necessarily resulting in an increase in the total number of requirements*

*Costello, R. and Liu, D. (1995). "Metrics for Requirements Engineering," Journal of Systems and Software. Vol 29 (No. 1), pp. 39-63*

# Requirements Volatility Trends (1 of 2)

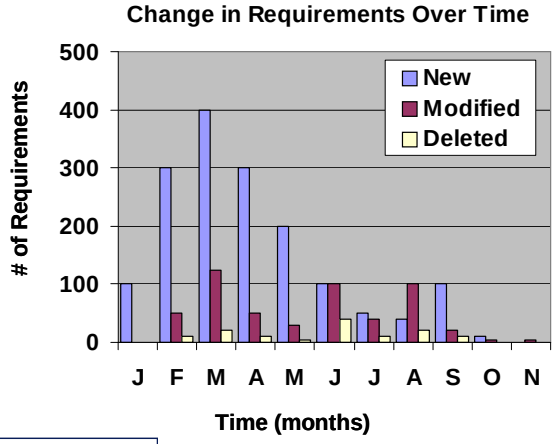



Notional Example





# Requirements Volatility Trends (2 of 2)




Notional Example

## 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 [Boehm 1991]
- 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 [Kotonya and Sommerville, 1995]
- 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
- System developers often lack effective methods and tools to account for and manage requirements volatility



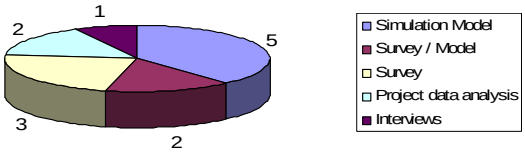


## Principal Research Question

**What technical, organizational, and contextual factors drive the amount of systems engineering effort added or reduced due the volatility of system requirements?**

# Literature Background

- Most of the requirements volatility research to date has been focused on software systems
- Various research methods have been utilized to investigate the causes and effects of requirement volatility – a methodological breakdown of the studies reviewed to date is below




- However, there still a lack of empirical data to determine the quantitative impact of requirements volatility on systems engineering effort for a broader base of engineering projects

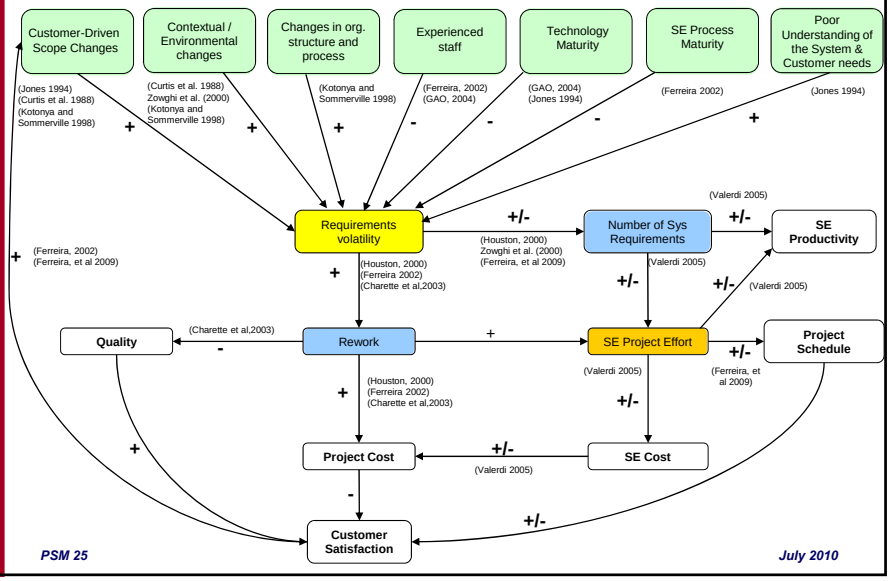




# Observations from the literature

1. Requirements volatility is correlated with an increase in project size and systems engineering effort
2. Requirements added after SRR have a greater impact on effort than requirements of comparable complexity captured in the initial baseline
3. The level of volatility in the requirements set is a function of the system life cycle phase
4. The impact of adding, modifying, or deleting a requirement increases the later the change occurs in the system lifecycle
5. Removing a requirement may not necessarily result in a net decrease in systems engineering effort
6. Based on the literature, a causal model was developed that relates technical, organizational and contextual project factors to requirements volatility

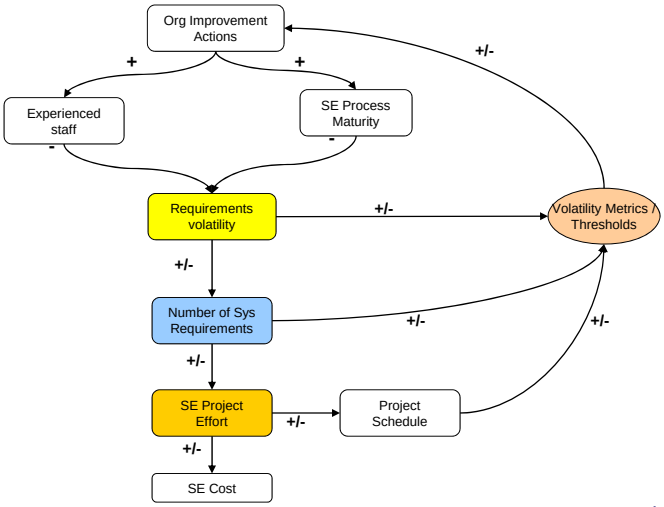
# Causal Model (normative)








# Moderating impact of expected volatility & thresholds



# Questions for discussion

1. Are there other important causes of volatility missing in the model?
2. Do you agree with the polarity of the relationships?
3. In what cases is the relationship between volatility and # of systems requirements a positive one, and in what cases is it a negative one?
4. Should the impact of requirements volatility be adjusted based on the criticality/coupling of the requirements?
5. Does volatility have an impact on productivity?
6. Should volatility thresholds vary depending on the size and duration of a project?





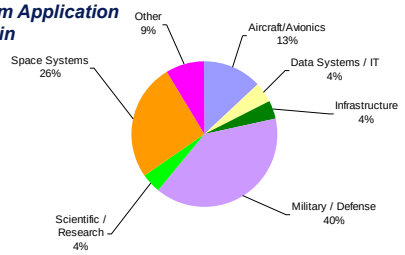
# Exploratory Survey

- An exploratory survey was developed to gather the perspectives of subject matter experts on the causes, impacts, and expected level of requirements volatility for a given system of interest
- The survey was piloted during the 2010 USC-CSSE Annual Research Review
- Version 2.0 of the survey was administered at the 2010 LAI Knowledge Exchange Event
- Organizations represented included:
  - The Aerospace Corporation, Northrop Grumman Corporation
  - The Boeing Company, Softstar Systems, Raytheon
  - United Launch Alliance, Massachusetts Institute of Technology, University of Southern California, and
  - Representatives from the United States Army and Navy

## Practical Software and Systems Measurement

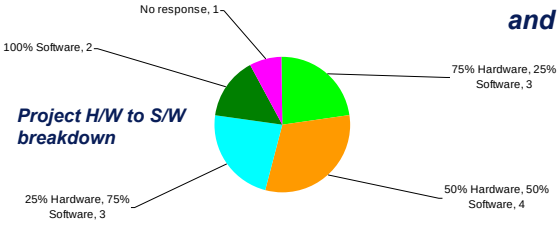
# USC-CSSE Annual Research Review Participants Background

**System Application Domain**




- *24 years average industry experience*
- *Primarily from a Military/Defense and Space Systems Background*
- *Experienced on Systems with a fairly balanced H/W and S/W work content*

**Project H/W to S/W breakdown**

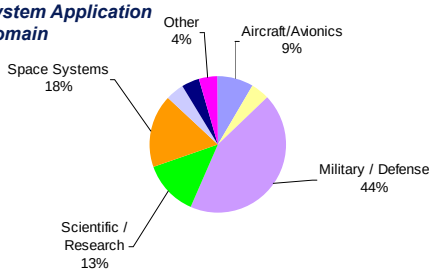





## Practical Software and Systems Measurement

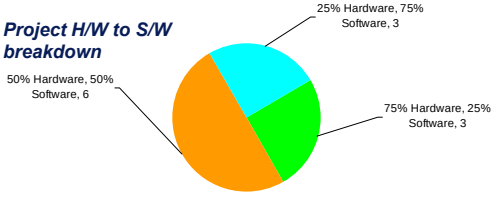
# LAI Knowledge Exchange Event Participants Background

**System Application Domain**




- *22 years average industry experience*
- *Primarily from a Military/Defense and Space Systems Background*
- *Experienced on Systems with a fairly balanced H/W and S/W work content*

**Project H/W to S/W breakdown**



## Summary of Survey Results:
### Use of Requirements Volatility Metrics

- Most participants either agreed or strongly agreed that requirements volatility metrics enable them to monitor and improve the performance of their project (46% USC ARR, 82% LAI)
- However, a sizeable percentage responded that their organizations do not use requirements volatility metrics (36% USC ARR, 63% LAI)
- There seems to be a disconnect between individual contributors' perspectives and organizational adoption of requirements volatility metrics

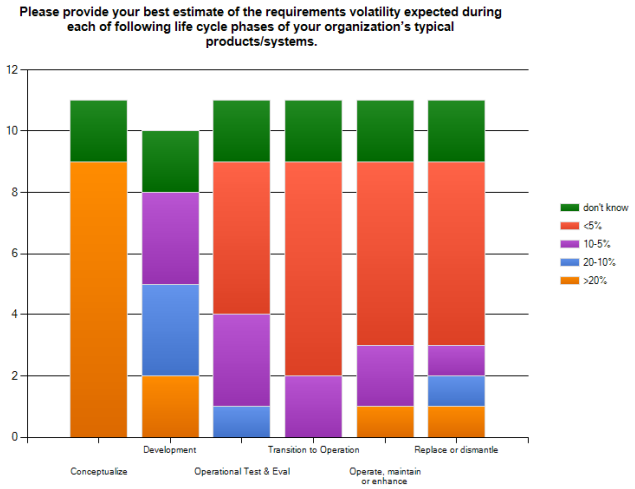




## Summary of Survey Results:
### Expected Levels of Volatility

- Most respondents expect >10% volatility during the conceptualize and development phase of the project, and <10% volatility for the rest of the system life cycle
- Participants who work on software-intensive systems expect a higher level of volatility in the later stages of the project than respondents from hardware oriented systems
- Most survey participants stated that the type of project (experimental, development, production, etc.) has a high to very high influence on the expected level of requirements volatility

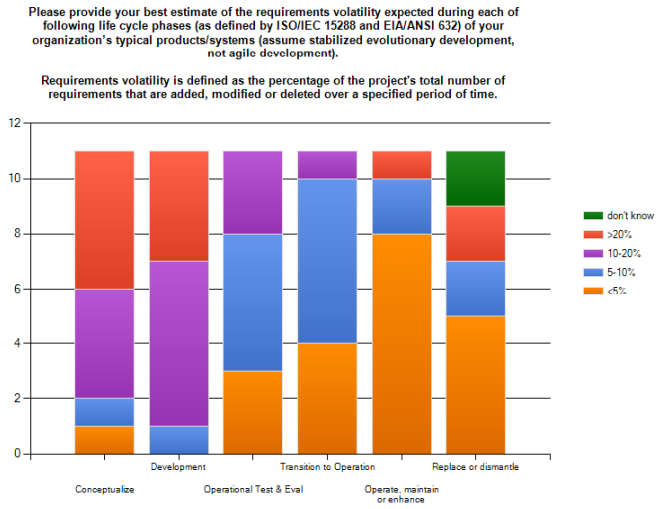
# Expected Volatility:
## USC-CSSE ARR Survey



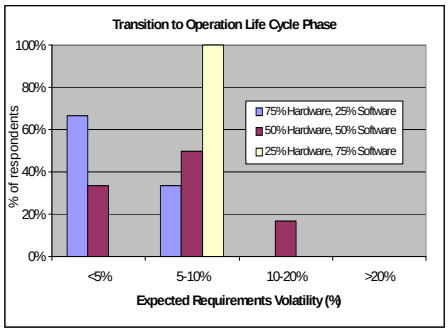


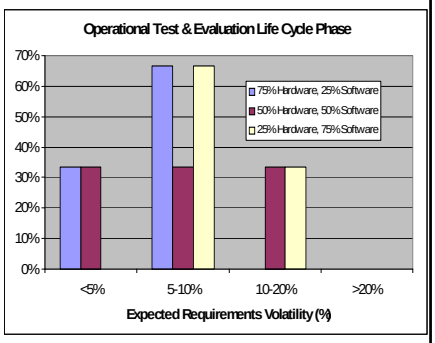



# Expected Volatility:
## LAI Knowledge Exchange Survey



# Impact of Hardware/Software Project Breakdown on Expected Volatility




***Respondents that work on projects with 75% S/W content expect a higher level of requirements volatility in the test through operational life cycle phases than respondents with projects with 75% H/W content***



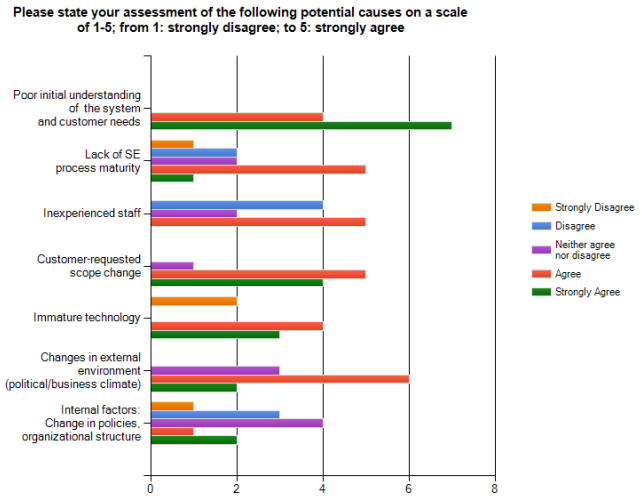


# Summary of Survey Results:
## Causes and Impacts of Volatility

- ***In general, preliminary results of the survey support observations from the literature and causal model***
  - ***Most respondents stated that requirements volatility will cause a moderate to large increase in the number of system requirements and the amount of rework***
- ***There were additional findings with respect to the strength of the relationship between variables:***
  - ***All respondents either agreed or strongly agreed that "Poor initial understanding of system requirements or customer needs" is a cause of requirements volatility***
  - ***"Changes in organizational structure and policies" had the lowest level of agreement as a cause of requirements volatility***

# Causes of Volatility

## USC-CSSE ARR Survey

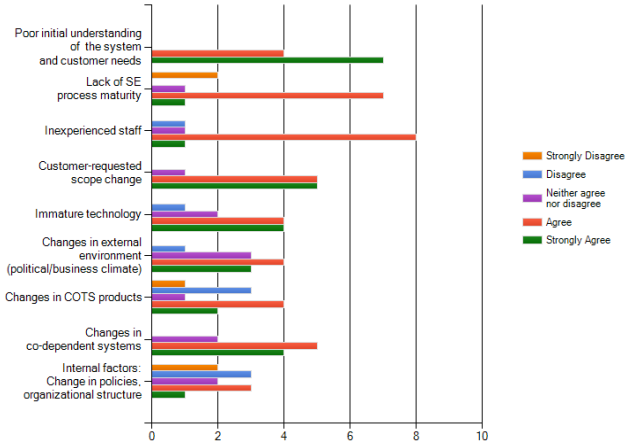






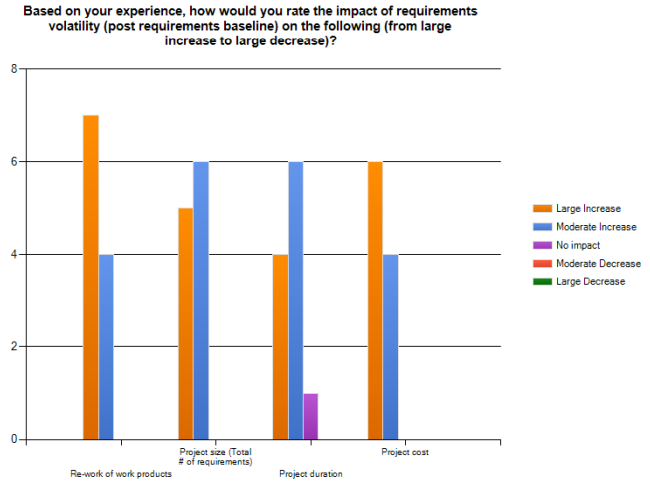

# Causes of Volatility

## LAI Knowledge Exchange Survey



# Impacts of Volatility

## USC-CSSE ARR Survey

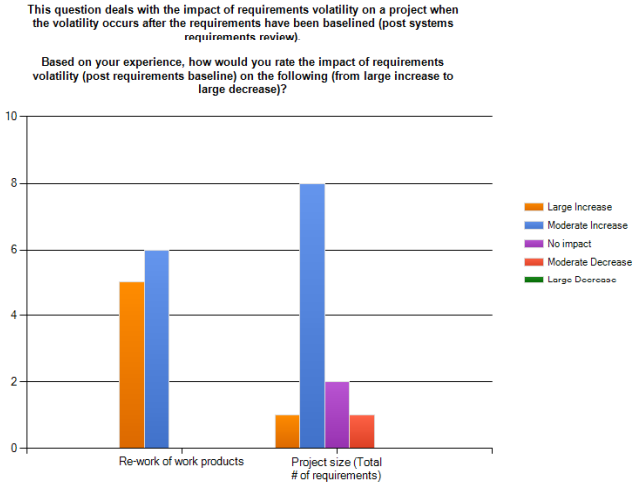




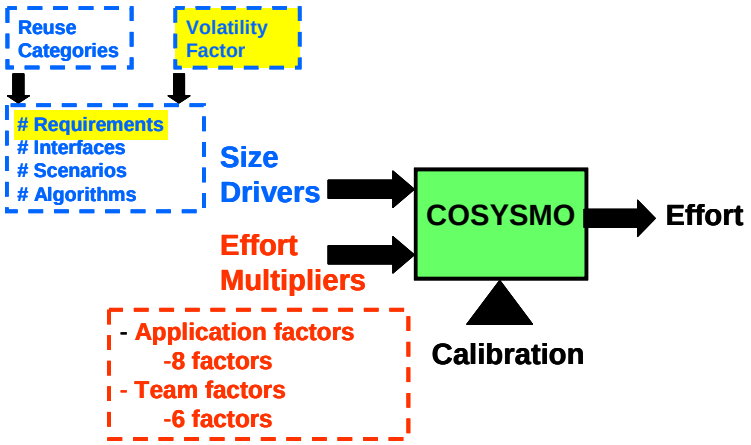



# Impacts of Volatility

## LAI Knowledge Exchange Survey



# Implications to COSYSMO








# Proposed COSYSMO Extension

- *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 model*
- *One of the objectives of the research is to complete the requirements volatility extension to COSYSMO within the existing structure and scope of the model*
- *The proposed extension builds upon the COCOMO II method of using a size adjustment factor to account for Requirements Evolution and Volatility (REVL)*

*Boehm, B., Abts, C., Brown, A.W., Chulani, S., Clark, B., Horowitz, E., Madachy, R., Reifer, D.J., and Steece, B. (2000). Software Cost Estimation with COCOMO II. Prentice Hall.*

# Volatility Adjustment Factor (1 of 3)

REVL is defined as the percentage of the baseline set of requirements that is likely to change due to the technical and organizational factors captured in the causal model

This relationship is expressed through the following equation:

$$R_{eff} = \left(1 + \frac{REVL}{100}\right) \times \left(R_0\right)$$

Where,

$R_0$ = Baseline number of requirements

$R_{eff}$ = Effective number of requirements at the end of the project

The effective increase in the number of requirements would result in an associated increase in systems engineering effort

Boehm, B., Abts, C., Brown, A.W., Chulani, S., Clark, B., Horowitz, E., Madachy, R., Reifer, D.J., and Steece, B. (2000). Software Cost Estimation with COCOMO II. Prentice Hall.





# Volatility Adjustment Factor (2 of 3)

In COSYSMO, the requirements are categorized by level of complexity as "easy," "nominal," and "difficult"

Applying the three categories to the equation below results in the following relationship

$$R_{eff} = \left(1 + \frac{REVL}{100}\right) \times \left(R_{e,r} + R_{n,r} + R_{d,r}\right)$$

Where,

$R_{e,r}$ = Initial number of requirements classified as "easy"

$R_{n,r}$ = Initial number of requirements classified as "nominal"

$R_{d,r}$ = Initial number of requirements classified as "difficult"

# Volatility Adjustment Factor (3 of 3)

*Observations from the literature indicate that requirements added post-SRR carry an effort penalty due to the potential rework and collateral impact to other engineering products*

*A weighting factor is added to account for this additional effort by increasing the effective functional size of the project*

$$R_{eff''} = w_v\left(1+\frac{REVL}{100}\right)\times\left(R_{e,r}+R_{n,r}+R_{d,r}\right)$$

*Where,*

$w_v$ = *Requirements volatility weighting factor*





# Proposed Revised Algorithm

$$\Phi_r = \left(\sum_v w_v \cdot \left(1+\frac{REVL}{100}\right)\left(\sum_r w_r\left(w_{e,r}\Phi_{e,r}+w_{n,r}\Phi_{n,r}+w_{d,r}\Phi_{d,r}\right)\right)\right)$$

*Where,*

$\Phi_r$ = *total quantity of the requirements size driver*

*REVL = Requirements Volatility and Evolution Factor*

$w_{xr}$ = *weight for "Easy", "Nominal", or "Difficult" size driver*

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

$w_r$ = *weight for reuse category*

$w_v$ = *Requirements volatility weighting factor*

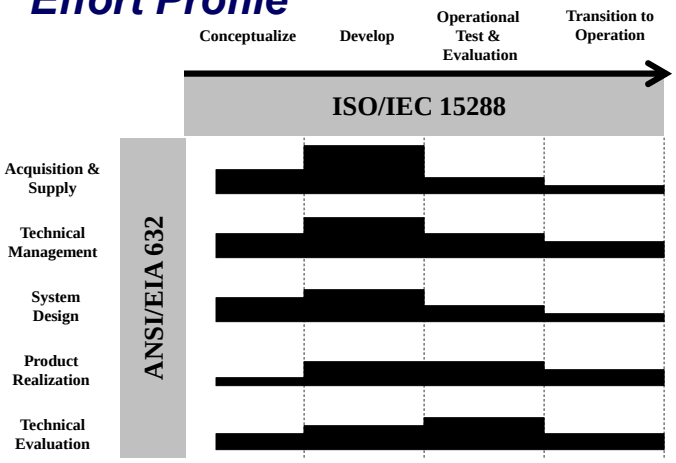
# Use Case: Accounting for Requirements Volatility

- ***Goal:*** *Account for the impact of requirements volatility on systems engineering effort for a given system of interest*
- ***Summary:*** *Changes the requirements set are expected to increase the functional size of the project and cause rework, which has an impact on systems engineering effort*
- ***Actors:*** *Systems Engineer, project manager*
- ***Components:*** *Original COSYSMO algorithm, proposed algorithm extension, REVL estimate, requirements volatility weighting factor*
- ***Normal Flow:***
  1. *Enter size parameters for the system of interest*
  2. *Enter reuse information if applicable*
  3. *Enter Requirements Evolution and Volatility (REVL) factor*
  4. *Select cost parameters for system of interest*
  5. *COSYSMO Extension Outputs*
     - *Systems Engineering Person Months*





# COSYSMO Systems Engineering Effort Profile



# Life Cycle Phase Definition

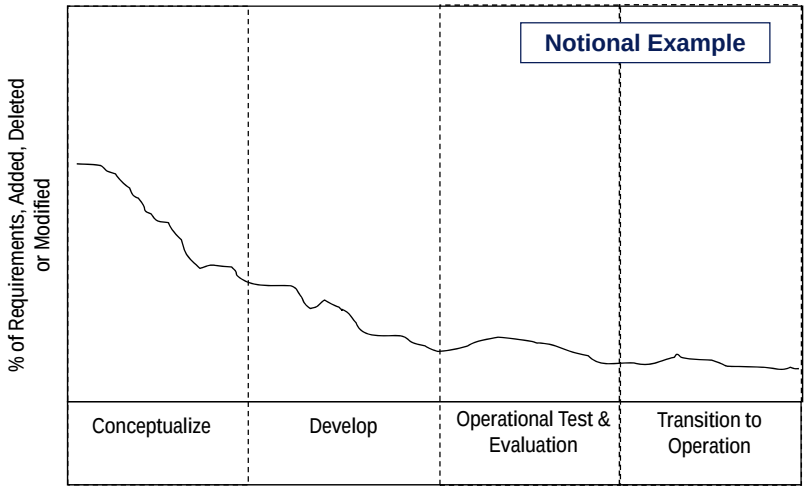
- ***Conceptualize stage*** *focuses on identifying stakeholder needs, exploring different solution concepts, and proposing candidate solutions.*
- ***The Development stage*** *involves refining the system requirements, creating a solution description, and building a system.*
- ***The Operational Test & Evaluation stage*** *involves verifying/validating the system and performing the appropriate inspections before it is delivered to the user.*
- ***The Transition to Operation stage*** *involves the transition to utilization of the system to satisfy the users' needs.*

*Valerdi, R. (2005). The constructive systems engineering cost model (COSYSMO). Doctoral Dissertation. University of Southern California, Industrial and Systems Engineering Department.*

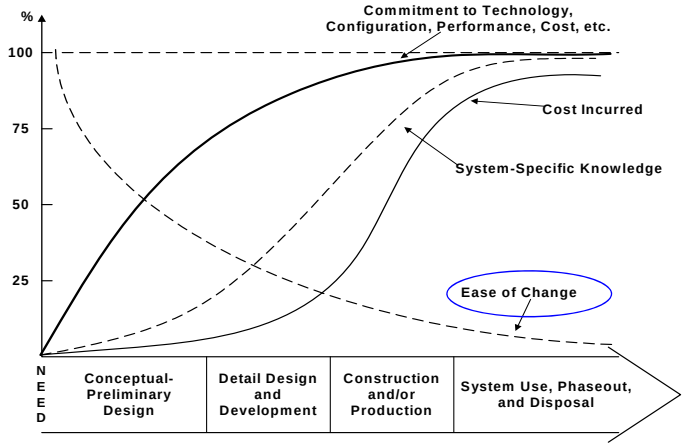




# Determine Expected Requirements Volatility Profile



# Cost Commitment on Projects

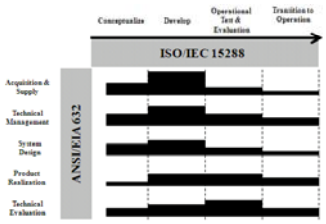



*Blanchard, B., Fabrycky, W., Systems Engineering & Analysis, Prentice Hall, 1998.*

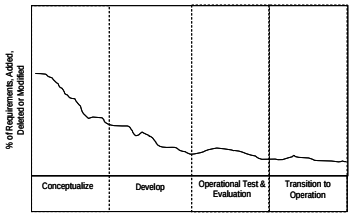


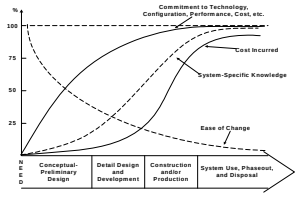


# Aggregated SE Effort Profile




X




*Requirements volatility weighting factor = 1 / ease of change*

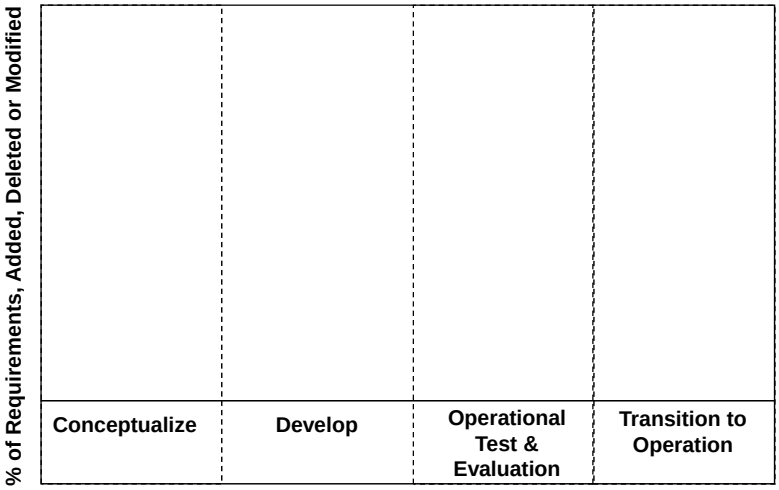
# Survey Exercise

1. Draw requirements volatility profile across the lifecycle phases covered by COSYSMO
2. Draw the "ease of change" profile across the same life cycle phases to determine the volatility weighting factor
3. Discuss variation in 1 and 2 above for
   1. Large and Small Projects
   2. Hardware and Software Projects
   3. Development and Recurring Projects

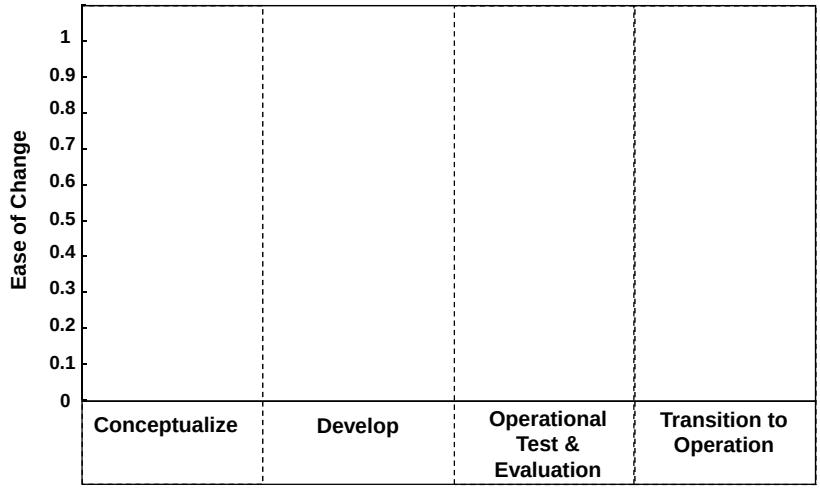




# Determine Expected Requirements Volatility Profile



# Determine Volatility Weighting Factor




*Requirements volatility weighting factor = 1/ ease of change*





# 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