Integration of Parametric Cost Estimation with System Architecture

... and How It Applies to SE Productivity Metrics

Mr. Barry Papke, No Magic
Dr. Gan Wang, BAE Systems

19th PSM Users’ Group Conference
11-13 August 2018
Arlington, VA
Measuring Productivity: Terminology

- **A General Production Model:**
  \[ Q = f(K, L) \]
  where,
  \( Q \) = output; \( K \) = capital; \( L \) = Labor

- **Productivity,** in general economic terms:
  \[ \text{Productivity} = \frac{\text{Output Created}}{\text{Input Used}} \]

- **Labor productivity** (LP) is typically measured as output per worker or *output per labor-hour*.
  \[ LP = \frac{\partial Q(K, L)}{\partial L} \]
Systems Engineering Productivity

- We have established:

- **SE Productivity**: Productivity for systems engineering is defined as the amount of the system (measured in eReq) produced or realized per unit of labor (measured in eng. hour)

  \[
  SE \text{ Productivity} = \frac{System \ Size}{Total \ SE \ Hours} \quad (eReqs/SE \ Hours)
  \]

- **SE Efficiency**: Efficiency for systems engineering is defined as the number work hours or effort (measured in eng. hours) required to produce a given unit of system (measured in eReq)

  \[
  SE \text{ Efficiency} = \frac{Total \ SE \ Hours}{System \ Size} \quad (SE \ Hours/eReq)
  \]
SE Productivity Measure Is Based on COSYSMO CER

4 Size Drivers and 14 Cost Drivers:

- Application factors (8 factors)
- Team factors (6 factors)

Size Drivers

Effort Multipliers

Effort

Calibration (Historical Data)
**Normalized Systems Engineering Productivity**

- **Normalized SE Productivity**: amount of the system produced or realized per unit of labor, under the *nominal* system complexity and project environment
  - Mathematically

\[
SE\ Productivity_{Norm} = SE\ Productivity \cdot CEM = \left(\frac{System\ Size}{Total\ SE\ Hours}\right) \cdot CEM
\]

Where,

\[
CEM = \text{the composite effort multiplier defined from 14 cost drivers}
\]

\[
CEM = \left(\prod_{i=1}^{8} AF_i\right)^{1/8} \cdot \left(\prod_{j=1}^{6} TF_j\right)^{1/6}
\]

(COSYSMO 3.0 CER)

- “Normalize” different projects to the same level of complexity – or, take the complexity/environment “out of the equation”
Question 1/2: What to Measure?

- We have also established:

- COSYSMO 3.0:
  - Size Drivers = {REQ, IF, ALG, SCN}

  - Levels of Complexity
    - “Easy”
    - “Nominal”
    - “Difficult”

  - Degrees of Reuse
    - “Generalized Reuse Framework”

  ![Graph showing complexity levels and system drivers](image-url)
Question 2/2: How to Measure?

- We have further established: the linkage between COSYSMO size drivers and SysML based architecture attributes

Note: the principle applies to all modeling tools
What Has Established...

... System Models Provide Direct Estimating Size Driver Inputs

Each weighted by:
1) Levels of complexity
2) Categories of reuse

Our Motivation: Integrated System Design and Cost Estimation

"Single Source of Truth" – Extending the digital thread into the cost domain!
A New SE CONOPS – Estimating as an Integral Part of System Architecture Modeling

Four (4) Integrated Modeling Activities:

Activity 1 – Create COSYSMO Profile (one-time configuration)
- SysML Profile Package
- DWR, DFR and Complexity Stereotypes and Counting Metrics
- Create once and reuse in any project

Activity 2 – Develop System Architecture
- Unchanged from current practice
- Application of modeling guidelines ensure consistent definition across projects

Activity 3 – Identify Cost Model Inputs
- Enabled by tool automation and SysML constructs
- Assigns Reuse Category and Complexity

Activity 4 – Generate Estimate
- Execution of the cost model is unchanged
- Cost model inputs are now part of the system architecture definition

These Activities Are Performed as Part of an Augmented System Modeling Process

Activities 2 & 3: we do this already, sort of...

Activities 1 & 4: This is what’s new!

*Connecting the digital thread into the cost estimation domain!*
Activity 1: Create COSYSMO Profile (one-time event)

- **SysML Profile** package enables creation of stereotypes for
  - Reuse categories: DWR and DFR
  - Level of complexity: easy, nominal, difficult
- Once created, it can be reused in any new model
Activity 2: Develop System Architecture

Development of the SysML architecture model is unchanged*

* Projects must adopt consistent modeling standards to ensure repeatable sizing estimates that are consistent with calibration data.
Activity 3: Identify Cost Model Inputs

Iteratively with Step 2...

Identify Sizing Elements

System of Interest (Count at the SOI Black Box Level):
- Requirements
- Interfaces
- Algorithms
- Operational Scenarios

Determine Reuse Category and Complexity

DFR
DWR
Level of Complexity

Count Sizing Elements (Automatic)

MBSE Tool features automate counting and collection of sizing data
A potential of 132 individual pieces of sizing data:
- 5 DFR Reuse Categories
- 6 DWR Reuse Categories
- 3 Levels of Complexity
- 4 Sizing Elements Types (REQ, ALG, SCN and IF)

Generate Cost Drivers

14 Cost Drivers:
- Qualitative System characteristics Assessed based on Behavior and Parametric Model Elements

Review for Consistency

The key to accurate and repeatable parametric cost estimation
Tool features enable detailed review and analysis of sizing data

Requires System Domain Experience

Rely on tool automation

Rely on SysML language properties

© 2018 BAE Systems and No Magic
Review Data for Consistency

Advanced query features enable comprehensive analysis of sizing inputs.

Example:
- The requirement that drove a specific critical algorithm should have similar DWR/DFR and complexity values as the SysML Activity that satisfies it.

Cross-cutting views and other analysis features of the MBSE toolset enable detailed review and analysis of sizing & cost driving parameters.
Integrated Modeling-Estimating Environment Enables Rapid Design Iteration and Optimization

- Sizing Data is a property of the architecture and maintained with the system model
- Alternatives can be quickly evaluated to achieve optimized design that meets:
  - Functional and Performance Requirements
  - Cost Targets
- Cost impacts can now be integrated into the systems engineering decision process
MBSE Allows Systems Engineers to Focus on the Important Things...

- **Tasks for MBSE Toolset**
  - Maintain Sizing Data as part of the System Architecture
  - Provide efficient User Interface to apply Sizing Parameters
  - Automate Counting
  - Provide Cross Cutting Views for Analysis

- **Tasks for the Systems Engineer**
  - Design the System
  - Determine Sizing Elements
  - Determine Reuse Category and Complexity
  - Analyze Results
Managing SE Productivity Metrics for Enterprise

- Combined model library with productivity metrics database
- Lifecycle metrics across product lines and organizations

The Prospect…

- As an attribute, SE productivity metric is embedded in system architecture models
  - Enabling systematic reuse and early design decisions
  - Enabling estimating capabilities: *analogy, parametric*
  - Connecting system (functions and performance) to economics
Conclusion with Perceived Benefits

- Integration of cost estimation with system modeling further extends the “digital thread”
  - Complete traceability from design to cost
  - **Repeatable** estimating with direct analysis/trade features
- Formalized development, integration, curation, and use of models for life cycle
  - **Early** system understanding
  - Reduced **cycle time** from design to cost, enabling to **earlier decision** making and **faster** time to market
- Enduring and authoritative “**Single source of truth**”
  - Reliable, trustworthy, and authoritative
  - Ultimately, better systems
Future Work

- Evaluation of tool-tool data exchange formats and protocols between SysML modelling and COSYSMO estimating tools
  - Potential MagicDraw add-on/extension
- Lifecycle management of cost estimation data within the MBSE repository as a corporate asset
- Conduct of one or more pilot case study projects
About the Authors

Barry Papke is the Director of Professional Services for No Magic Inc. He has thirty-two years of systems engineering and operations analysis experience in the aerospace and defense industry across the entire systems engineering lifecycle from concept development through integration, test and post-delivery support.

Gan Wang, Ph.D., is a Global Engineering Fellow at BAE Systems and the Chief Engineer for its Integrated Defense Solutions businesses. He has been actively engaged in systems engineering processes, cost estimating and analysis, modeling & simulation, multi-criteria decision making methods, and system-of-systems engineering methodologies.
Thank You

Dr. Gan Wang, BAE Systems
Mr. Barry Papke, No Magic