International Council on Systems Engineering
A better world through a systems approach

A Maturity and Cost Model for Systems Engineering with Generative AI

Ray Madachy
Naval Postgraduate School

Ryan Bell, Ryan Longshore
Naval Information Warfare Center and Naval Postgraduate School

Introduction

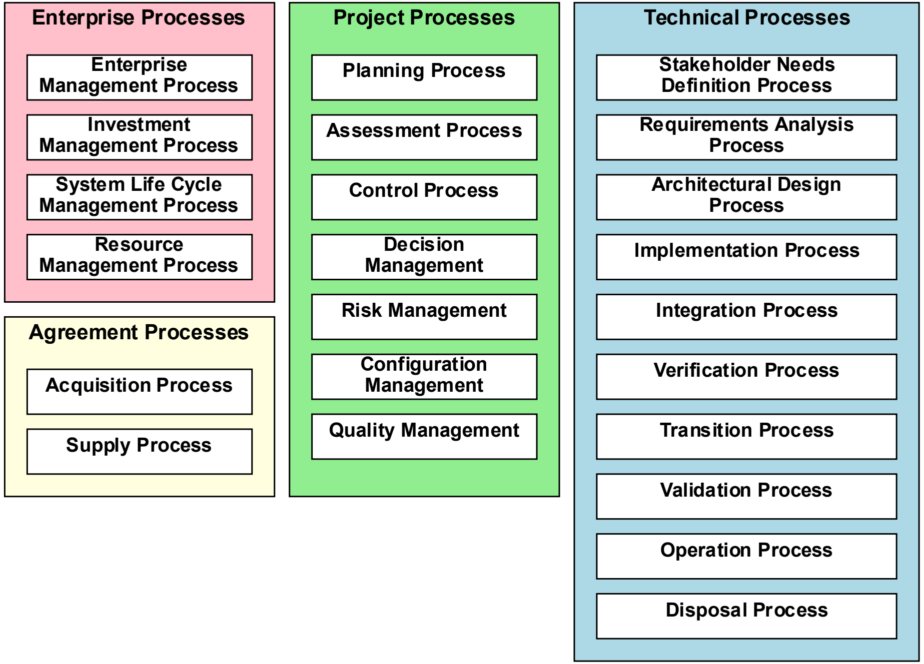
- Generative AI assistance is quickly transforming systems engineering processes
  - AI tools with large language models (LLMs) enable task automation, architecture generation, design, trade-off analysis, implementation, testing, and more.
  - Supports decision-making, risk management, and continuous process improvement.
  - Drastic reductions in effort when applied effectively
- Addressing new challenges in disrupted cost models for systems and software engineering
- Why this framework?
  - Provides a structured approach for organizations to integrate AI effectively and optimize engineering processes.
  - Harmonizes AI adoption maturity and costs models
  - Aligns AI usage with quantitative cost impacts to better inform decision-makers.

Framework Overview

- Harmonized maturity and cost models [12].
  - Maturity model provides a roadmap for AI adoption, identifying key practices with five progressive levels.
  - Parametric cost model quantifies the cost impact of AI usage on systems engineering projects.
  - Underlying ISO/IEC/IEEE15288 [6] harmonization of systems and software engineering activities.
- Introduction of AI Usage factor for cost modeling using Constructive Cost Model (COCOMO) [8] and Constructive Systems Engineering (COSYSMO) [7] framework.
- Empirical data collection and process for cost model calibration using the maturity levels and cost factor rating scale for AI Usage.
- Cost data also aligned with practices and activities in the ISO/IEC/IEEE15288 lifecycle standard [6].
- Cost modeling improves predictive accuracy for tradeoff analysis, project estimation, budgeting, and process benchmarking to facilitate continuous improvement of AI adoption.

What Costs?

- Labor cost of teams performing systems engineering processes.
- Cost of producing system requirements, system interfaces, critical algorithms, operational scenarios, and supporting artifacts.
- Process phases and activities aligned with the ISO/IEC/IEEE 15288 standard.



AI Usage Potential Cost Decreases

Activity	Benefits
Acquisition and Supply	Assist in drafting acquisition strategy, evaluating supplier proposal, and generating contract requirement traceability. Enable rapid review of past contract and risk clause.
Technical Management	Support planning by generating work breakdown structure, risk register, and configuration guideline. Monitor project data to alert deviation from performance target.
System Design	Recommend design pattern, generate architecture option summary, and propose interface concept. Generate models, and assist in trade analysis using prior design rationale.
Product Realization	Help create specification, checklist, and realization workflow. Monitor manufacturing readiness data or digital twin output for deviation.
Product Evaluation	Suggest test objective, derive evaluation plan, and summarize finding from past verification activity. Assist in validation scenario generation and mission alignment check.

AI Usage Potential Cost Increases

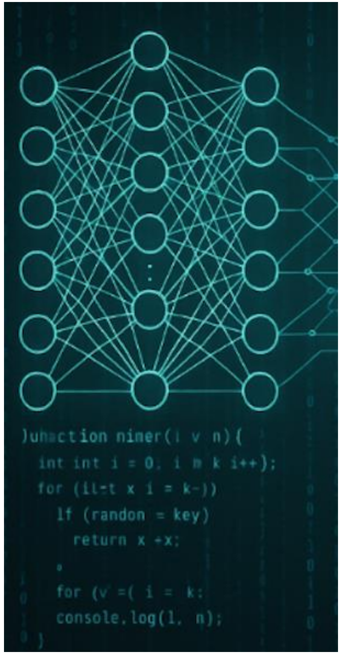
Activity	Downfalls
Acquisition and Supply	Relying on AI to evaluate supplier or contract artifact without expert review might overlook contextual risk or misinterpret requirement intent.
Technical Management	Overuse of AI for planning or risk identification might produce false confidence in completeness or cause oversight of emergent issue.
System Design	Accepting architecture suggestion without human vetting can lead to brittle or non-viable solution in complex or novel domain.
Product Realization	Using AI-generated specification or workflow blindly can introduce integration error or unverified assumption in production step.
Product Evaluation	AI-generated test or validation scenario might miss edge case or domain-specific constraint if not critically reviewed by engineer.

Maturity Model Overview

- Organizations progress through levels by upgrading infrastructure, data management, workforce training, and governance.
  - Robust data management and access infrastructure are critical for supporting generative AI tools, which rely on large datasets for training and inference.
  - Engineers must be trained to effectively use generative AI tools. This includes not only technical skills but also an understanding of AI limitations and how to interpret AI-generated outputs.
  - Successful adoption requires rethinking engineering workflows to fully integrate AI-generated insights, ensuring that AI contributes meaningfully to systems engineering goals.
  - Clear policies must be established regarding the use of AI in engineering, including issues of accountability, transparency, and bias mitigation.
- Maturity levels are defined for implementing key practices
  - Level 1: Initial - Minimal AI adoption, mostly traditional engineering workflows.
  - Level 2: Developing - AI usage in select tasks like documentation, basic automation.
  - Level 3: Competent - AI regularly used in design, modeling, and analysis.
  - Level 4: Advanced - AI plays a strategic role in decision-making and optimization.
  - Level 5: Optimized - AI is fully integrated, automating most engineering workflows.

Key Practice Terminology

- Large Language Models (LLMs): Neural networks to understand and generate text and other forms of data.
- Training: Teaching a model from scratch how to perform a task using large amounts of data.
- Retraining: Retraining a base model with a new dataset.
- Finetuning: Teaching a base model specialized knowledge by adjusting internal parameters such as weights and biases.
- Retrieval Augmented Generation (RAG): Utilizing external data to augment an LLM’s knowledge without changing model parameters.
- Reinforcement Learning from Human Feedback (RLHF): An iterative technique used to train AI models by incorporating human feedback to improve their performance and alignment with human preferences.



Maturity Model Key Practices

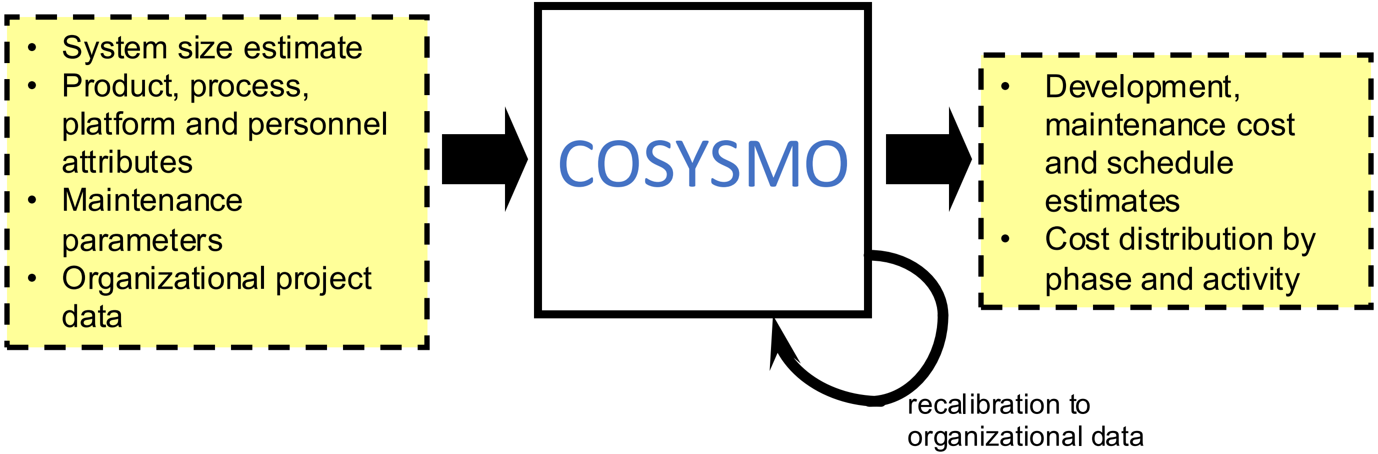
- **Investment in Hardware and Infrastructure** addresses the technological resources required for AI adoption, ranging from basic infrastructure to highly scalable, robust systems.
- **Knowledge Integration and Data Management** highlights how AI systems evolve from limited context awareness to fully indexed and dynamically managed knowledge bases.
- **LLM Customization and Fine-Tuning** describes the depth of AI model customization, from using a generic model to extensive fine-tuning with specialized datasets.
- **LLM Retraining** indicates the frequency and adaptiveness of model updates, evolving from minimal retraining to real-time adaptive learning with Reinforcement Learning from Human Feedback (RLHF).
- **Personnel Training** captures the progressive development of workforce competencies to improve AI-assisted engineering processes. Without adequate training, organizations risk misuse, over-reliance, or underutilization of AI tools, leading to inefficiencies or defective engineering outputs.

Maturity Levels

Practice	Level 1: Initial	Level 2: Developing	Level 3: Competent	Level 4: Advanced	Level 5: Optimized
Investment in Hardware and Infrastructure	Minimal, basic cloud subscription or server for API calls.	Low to moderate, added storage and processing for few-shot or limited RAG/fine-tuning.	Moderate, scalable storage and processing for an evolving knowledge base and RAG integration.	High, GPU-based infrastructure for extensive fine-tuning and large data processing.	Very high, robust infrastructure for multi-agent orchestration, real-time updates, and continuous learning.
Knowledge Integration and Data Management	Minimal, relying solely on model's inherent knowledge.	Basic, limited context-awareness using few-shot or small, static RAG knowledge base.	Moderate, using a fully indexed knowledge base to improve context-aware responses.	High, large, domain-specific knowledge base updated regularly for accuracy.	Real-time, cooperative agents with dynamic retrieval, high context sensitivity, and advanced data management.
LLM Customization and Fine-Tuning	None, using a generic pre-trained model without customization.	Minimal, few-shot learning or minor fine-tuning with small datasets.	Moderate, fine-tuning for specific tasks or domains with RAG integration.	High, extensive fine-tuning with proprietary or domain-specific data for customization.	Extensive, with specialized agents and continual fine-tuning for adaptability.
LLM Retraining	None, no retraining—uses only pre-trained model.	Basic periodic retraining with small data samples as needed.	Scheduled fine-tuning sessions based on new domain-specific data.	Continuous domain-specific retraining at intervals or after major data changes.	Real-time or adaptive retraining using RLHF and continuous updates with interaction.
Personnel Training	No formal training on AI tools; usage is ad hoc and based on personal initiative.	Limited training programs focused on basic AI tool usage and guidelines.	Standardized training on AI-assisted workflows, with emphasis on integrating AI into engineering tasks.	Advanced training programs incorporating best practices for AI-human collaboration and workflow optimization.	Continuous learning culture with AI literacy deeply embedded in engineering practices, with mentorship and adaptive training based on real-time AI evolution.

Cost Modeling

COSYSMO Black Box Model



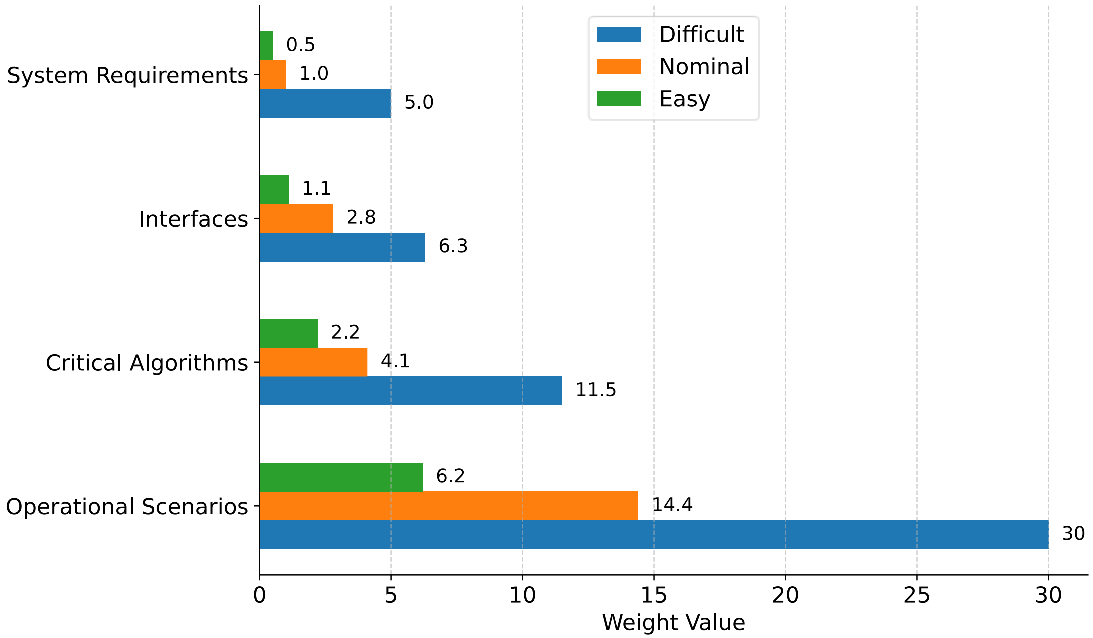
COSYSMO Effort Equation

$$Effort = A * Size^{B} * \prod_{i=1}^{N} EM_i$$

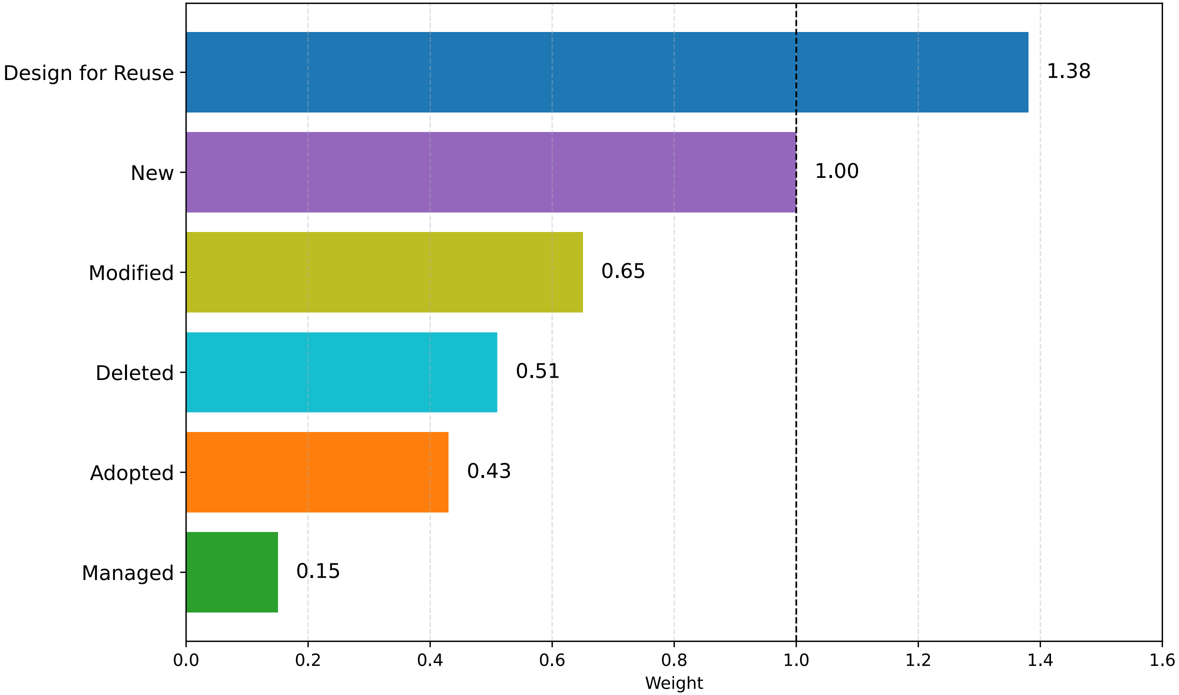
Where

- *Effort* is in Person-Months (PM)
- *A* is a constant derived from historical project data
- *Size* is a sum of weighted system requirements, interfaces, algorithms and scenarios
- *B* is an exponent for the diseconomy of scale
- EM_i is an effort multiplier for the i^{th} cost driver. The geometric product of *N* multipliers is an overall Effort Adjustment Factor (*EAF*) to the nominal effort.

Size Driver Weights



Size Driver Reuse Weights



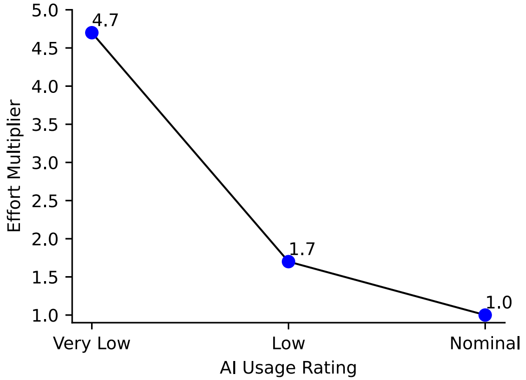
Cost Driver Rating Scales and Effort Multipliers

	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

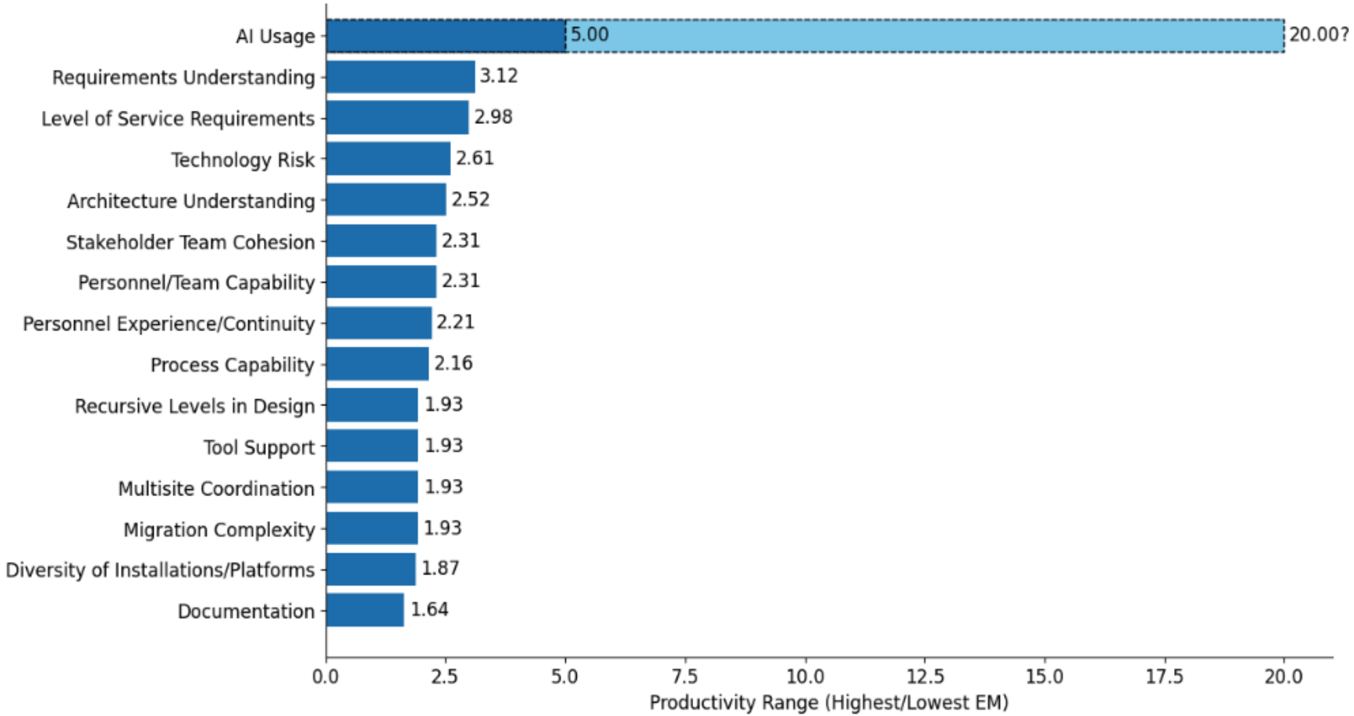
EMR = Effort Multiplier Ratio

AI Usage Project Cost Factor Ratings and Provisional Effort Multipliers

Very Low	Low	Nominal	High	Very High
Minimal to no AI assistance. Development relies on traditional methods and tools. AI tools may be present but are rarely, if ever, consulted.	Moderate AI usage, typically for summarization, basic information retrieval, and model templating. AI tools are not deeply integrated into the development workflow.	Regular use of AI tools for various tasks like MBSE help, design insights, or testing assistance. AI tools are a recognized part of the toolkit but aren't central to development	Frequent and strategic use of AI assistance. AI tools play a central role across the lifecycle from architecting, to design, implementation, and V&V.	AI tools are deeply ingrained in most activities. They are crucial for decision making, problem solving, and automating tasks. The development process is designed around maximizing AI tool benefits.



COSYSMO Cost Driver Productivity Ranges



Updated COSYSMO Tool

Constructive Systems Engineering Cost Model (COSYSMO)

System Size

	Easy	Nominal	Difficult
System Requirements	11	23	3
System Interfaces	2	6	5
Critical Algorithms	5	5	2
Operational_Scenarios	3	4	2

Cost Factors

Requirements Understanding	High	Documentation	Nominal	AI Usage	High
Architecture Understanding	Low	Diversity of Installations/Platforms	Nominal	Personnel Experience/Continuity	Nominal
Level of Service Requirements	Nominal	Recursive Levels in Design	Nominal	Process Capability	Nominal
Migration Complexity	Very High	Stakeholder Team Cohesion	High	Multisite Coordination	Nominal
Technology Risk	Nominal	Personnel/Team Capability	Nominal	Tool Support	Nominal

Labor Rate

Cost per Person-Month (Dollars) 20000

Calculate

COSYSMO with AI

- COSYSMO with AI: https://softwarecost.org/tools/COSYSMO_AI/ (or lowercase)
- Basic COSYSMO: https://softwarecost.org/tools/COSYSMO/
- System cost model suite with COSYSMO, COCOMO (software engineering), and hardware production: https://softwarecost.org/tools/cost_model_suite/

Example Investment Analysis

- A systems engineering organization is currently at maturity Level 2 (Developing) and wants to advance to Level 3 (Competent). They develop 1500 Equivalent Nominal System Requirements per year on projects at $20,000/Person-Month.
- It is desired to estimate the annual cost savings after AI adoption and break-even point when the investment will pay for itself.
- Investment costs to achieve Level 3 per maturity level practices:

Expense	Cost
All systems engineers take generative AI training	$300K
High performance computers purchased for LLM training	$150K
LLM team will retrain organizational models	$180K
Total Investment	$630K

- After adoption, their project practices will improve from the AI Usage rating from Low to Nominal.
- When is the break-even point?

Investment Analysis Results

- **As-Is (*AI Usage* = Low)**

  $Effort = A * Size^B * EAF$ = .054 * $1500^{1.06}$ * 1.7 = 213.6 PM

  Annual Cost = $20,000/PM * 213.6 PM = $4.27M

- **To-Be (*AI Usage* = Nominal)**

  $Effort = A * Size^B * EAF$ = .054 * $1500^{1.06}$ * 1.0 = 125.6 PM

  Annual Cost = $20,000/PM * 125.6 PM = $2.51M

- From the above, the annual savings and break-even point can be calculated:

  Annual Savings = $4.27M - $2.51M = $1.76M

  Break Even = Investment / Annual Savings = $.63M / $1.76M = .36 years

- The investment will pay for itself in slightly over 4 months of adoption on systems engineering projects.
- Thus, it would be a worthwhile investment, and the ROI would be a substantial multiplier after a few years.

Empirical Studies

Data Collection and Analysis Methods

- Multi-project data collection in conjunction with other cost factors, which was used to develop COSYSMO [7] and COCOMO II [8].
- Controlled group experiments comparing activities performed with and without AI assistance.
    - Naval Information Warfare Center (NIWC) systems engineering activities.
    - Public sources, e.g., GitHub study [10].
- Small-scale empirical case studies in the classroom and performed by researchers.
- Delphi surveys for expert judgment.
- Bayesian approaches combining empirical project data and Delphi results.

Online Data Form

- Submit at http://softwarecost.org/data/ai

AI Assistance Usage Ratings

Very Low	Low	Nominal	High	Very High
• Minimal to no AI assistance. • Development relies primarily on traditional methods and tools. • AI tools may be present but are rarely, if ever, consulted.	• Occasional AI consultation, typically for clarification or basic information retrieval. • AI tools are not deeply integrated into the development workflow.	• Regular use of AI tools for various tasks like code help, design insights, or testing assistance. • AI tools are a recognized part of the toolkit but aren't central to development.	• Frequent and strategic use of AI assistance. • AI tools play a central role in multiple phases of development, from design to code review.	• AI tools are deeply ingrained in most development phases. They are crucial for decision making, problem solving, and automating specific tasks. • The development process is designed around maximizing AI tool benefits.

Select only one method:

Method 1: Estimated Effort Multipliers

What are your estimated effort multipliers for each rating relative to Nominal set to 1.0?

1.0

Method 2: Estimated Effort Multiplier Ratio (EMR)

Provide your estimated effort ratio from Very Low to Very High. E.g., the overall EMR for *Applications Experience* is the ratio of 1.22/.81 = 1.5 for the multiplicative range. If you provide an effort ratio between two other settings then explain in Rationale and Supporting Information.

EMR =

Method 3: Effort Data from Project, Experiment or Case Study

Select the *AI Assistance Usage* rating applicable to the development: Nominal

Actual effort with AI assistance: Person-hours

Estimated effort without AI assistance: Person-hours

Please add a note if your effort without AI assistance is actual instead of estimated.

Rationale and Supporting Information:

Submit Delphi

Ideal Effort Multiplier

- Method to normalize out contaminating effects of other individual cost factors to isolate the contribution of the factor being analyzed on productivity.
- In our case, to analyze the contribution of AI Usage eliminating other cost factor sources of variance.

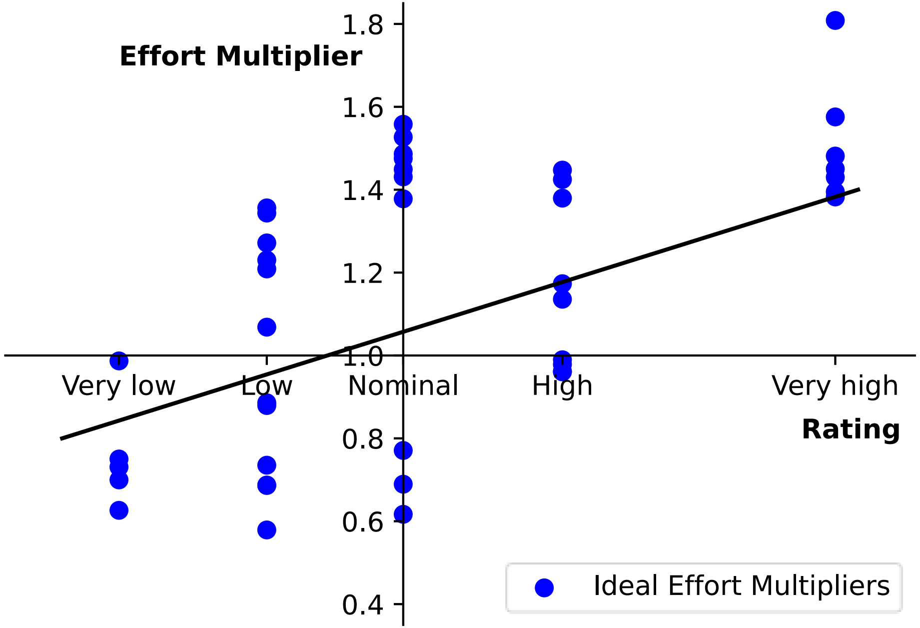
$$IEM(P, \text{Cost Factor}) = PM(P, actual)/PM(P, \text{Cost Factor})$$

where

- $IEM(P, \text{Cost Factor})$ is the ideal effort multiplier for project P
- $PM(P, actual)$ is the actual development effort of project P
- $PM(P, \text{Cost Factor})$ is the cost model estimate excluding the Cost Factor
- PM is Person-Months of effort

Ideal Effort Multiplier (Cont.)

- Click to add text



Conclusions and Future Work

- AI is transforming systems engineering, enabling efficiency, creativity, and better decision-making.
- Organizations must invest in training, infrastructure, and governance to optimize systems engineering processes with generative AI.
- The integrated maturity and cost modeling framework offers a structured pathway for organizations.
  - Serves as a roadmap for AI adoption, helping organizations understand where they currently stand and what steps are needed to advance.
  - Provides a basis for benchmarking progress and identifying best practices, fostering a culture of continuous improvement.
  - Cost model integration enables better informed cost and schedule estimates for projects using generative AI.
  - Supports holistic decision-making that considering technical capabilities, personnel skills, and and cost impacts.
- Future Work
  - Ongoing empirical calibration of cost models with real-world AI-assisted projects.
  - Refine the AI Usage cost factor based on ongoing industry feedback.
  - Develop open-source AI cost modeling tools.
  - Further analyze AI impacts across the lifecycle aligning artifacts and effort data with ISO/IEC/IEEE 15288.

References

1. Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. Capability Maturity Model for Software, Version 1.1. Software Engineering Institute, Carnegie Mellon University; 1993

2. CMMI Product Team. CMMI for Development, Version 1.3. Software Engineering Institute, Carnegie Mellon University; 2010.

3. Software Engineering Institute. Cybersecurity Maturity Model Certification (CMMC) 2.0. Software Engineering Institute, Carnegie Mellon University; 2021.

4. Madachy, R., Bell, R., Longshore, R. “Factoring AI Assistance into Software and Systems Engineering Cost Models”, Proceedings of the 2023 International Boehm COCOMO Forum; 2023.

5. Madachy, R., Bell, R., Longshore, R. "Systems Acquisition Cost Modeling Initiative for AI Assistance." Proceedings of the 2024 Acquisition Research Symposium, Monterey, CA.

6. International Organization for Standardization. ISO/IEC/IEEE 15288:2015 Systems engineering – System life cycle processes. International Organization for Standardization; 2015.

7. Valerdi, R. "The Constructive Systems Engineering Cost Model (COSYSMO)." PhD thesis, University of Southern California; 2005.

References (Cont.)

8. Boehm, B., Abts, C., Brown, W., Chulani, S., Clark, B., Horowitz, E., Madachy, R., Reifer, D., & Steece, B. Software Cost Estimation with COCOMO II. Prentice-Hall; 2000.

9. softwarecost.org. AI Assistance Usage Delphi Form, http://softwarecost.org/data/ai/, 2024.

10. GitHub. “Research: quantifying GitHub Copilot’s impact on developer productivity and happiness”. https:// github.blog/2022-09-07-researchquantifying-github-copilots-impact-on-developerproductivity-and-happiness, 2022.

11. Chulani, S., Boehm, B., & Steece, B. "Bayesian Analysis of Empirical Software Engineering Cost Models." IEEE Transactions on Software Engineering, 1999;25:4, pp. 573-583

12. Madachy, R., Bell, R., Longshore, R., “A Generative AI-Driven Systems Engineering Maturity and Cost Modeling Framework”, Proceedings of the 2025 Conference on Systems Engineering Research, Procedia Computer Science, 2025.