

PROJECT - PLAN B

AN INTRODUCTORY SIMULATION-BASED COMPARISON OF SYSTEMS
ENGINEERING FRAMEWORKS FOR MEDIUM-SCALE SCIENTIFIC INFRASTRUCTURE
PROJECTS

Submitted by

Hannah W. McLaurin

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Fall 2025

Master's Committee:

Advisor: Dr. Thomas Bradley

Dr. Steven Simske

Dr. Matthew Kipper

ABSTRACT

AN INTRODUCTORY SIMULATION-BASED COMPARISON OF SYSTEMS ENGINEERING FRAMEWORKS FOR MEDIUM-SCALE SCIENTIFIC INFRASTRUCTURE PROJECTS

Medium-scale scientific infrastructure projects exist between small laboratory experiments and large mission driven facilities. These projects face growing technical complexity and safety requirements while operating with limited resources and evolving goals. Traditional systems engineering frameworks such as the NASA Systems Engineering Handbook provide rigor but can burden research teams with documentation and governance unsuited to their agility. Minimal approaches allow flexibility but risk poor integration readiness and reduced data quality. CERN developed the OpenSE framework to offer a lean and safety focused alternative designed for research environments, yet its effectiveness outside accelerator contexts is not well studied.

This project compares three framework conditions: minimal approach, OpenSE, and traditional systems engineering. An initial, non-exhaustive literature review identifies success metrics for research infrastructure projects and establishes structural differences among the frameworks. A crude virtual project model, based on representative assumptions from experience working with the COHERENT experiment at Oak Ridge National Laboratory, is used to simulate project outcomes under each framework. Results provide an initial view of how framework choice influences project success and readiness in research infrastructure contexts.

ACKNOWLEDGEMENTS

I would like to thank Thomas Bradley and Ingrid Bridge for their support and ensuring that I both completed this project and created a quality product. I would like to thank Jason Newby, Brennan Hackett, and the COHERENT Collaboration for allowing me to not only be part of their team but use the incredible technology and physics instrumentation they designed and the hard work they put in to realizing these systems as a foundation for how to better understand the applications of systems engineering for fundamental physics research systems. I would also like to thank the engineers and physicists at CERN and NASA who created the frameworks being studied without which these questions would not be able to be posed.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
LIST OF TABLES	v
LIST OF FIGURES	vi
1 Introduction	1
1.1 Background and Motivation	1
1.2 Problem Statement	3
1.3 Purpose and Research Questions	3
1.3.1 Purpose	3
1.3.2 Project Scope	4
1.3.3 Project Questions	5
2 Literature Review	6
2.1 Framework Characteristics	6
2.1.1 NASA Systems Engineering Framework	6
2.1.2 OpenSE Framework	8
2.1.3 Minimal Frameworks	9
2.1.4 Comparative Summary	10
2.2 Advantages and Limitations of these Systems Engineering Frameworks	10
2.3 Research Infrastructure Project Metrics	13
2.4 Gaps in Simulation-Based Research	14
3 Methods	16
4 Results	19
5 Discussion	21
Bibliography	22

LIST OF TABLES

2.1	A Summary of Common Elements Across Frameworks	15
3.1	Simulation Variables and Descriptions	17
4.1	Summary of Statistics	20

LIST OF FIGURES

2.1	NASA Systems Engineering Framework [1].	7
2.2	OpenSE Lifecycle [2].	9
2.3	Relation of SE Investments to NASA Program Cost Overruns from SEBoK [3].	11
2.4	Risk-Balanced “How Much SE Is Enough” Plot from SEBoK [3].	12
4.1	Initial Scientific Yield Loss Monte Carlo Distribution	19

1 Introduction

1.1 Background and Motivation

Medium-scale physics research and scientific infrastructure projects occupy a unique space between small laboratory or bench-top experiments and large-scale facilities or missions. They are often demonstrators or used for benchmark measurements that are part of larger research programs. These programs are often led by multi-institutional collaborations exploring a specific scientific discovery. These collaboration-led projects are typically characterized by mid-scale budgets in the \$10 to 100 million range and often are attempting significant technical challenges that require a larger amount of research and development than large-scale projects [4].

As research programs mature, many collaborations experience a natural progression from these bench-top experimentation projects toward large, integrated system deployments. Early development stages often rely those teams' ability to be nimble and leverage ad hoc coordination, significant subject matter expertise, and reactive problem solving within small teams. However, as technical scope expands to include more instrumentation, increased complexity and scale of detectors, cross-institutional interfaces, and operational environment with increased safety critical functions, this informal model becomes insufficient. The increasing complexity and interdependence of components, including those between the instruments and the facility with which they reside, require higher levels of process discipline, documentation control, and configuration management. For many scientific teams, this transition represents a shift from research culture to systems engineering culture introducing practices that, while new to the community, are essential for ensuring integration readiness and sustained operational reliability. Not only are these collaborations now attempting to integrate complex scientific goals with engineering rigor but they are attempting to do so while operating under constrained resources and distributed management, thus driving the need for systems engineering as a risk mitigation in one form or another.

Across the scientific community, the practical applications of systems engineering vary widely. Standardized frameworks primarily originate from mission driven, high stakes, and highly regu-

lated environments that emphasize rigorous and documentation heavy systems engineering procedures. While these frameworks provide a robust foundation for disciplined system realization, their direct application to medium-scale research infrastructure projects is not well-defined or even well-accepted [5]. Scientific programs typically operate within flexible, evolving, and resource limited contexts that do not align with the formal rigor and requirements of aerospace, nuclear, accelerator, defense, etc. systems engineering frameworks. Moreover, even the tailoring of these systems engineering frameworks for medium-scale research infrastructure projects is perceived by the academic and research community as non-viable [5].

Assuming an adoption of systems engineering for a medium-scale research infrastructure project, framework selection thus becomes a critical determinant of project performance. Overly prescriptive frameworks can inhibit agility, decrease innovation, slow progress, and potentially increase wasted resources due to over-burdensome documentation; while insufficiently structured ones can obscure accountability, weaken integration readiness, and increase the likelihood of design rework and loss of meaningful scientific data.

As a response to these contextual constraints, CERN developed the OpenSE framework [2] to provide an alternative designed specifically for research projects that integrate into facilities with safety critical operations, (e.g., accelerators, nuclear reactors, etc.). It emphasizes safety, iterative development, and lean documentation practices while maintaining lifecycle alignment with recognized international standards. OpenSE aims to preserve the systems thinking discipline of traditional systems engineering frameworks while improving accessibility and adaptability for research driven environments. For context, within this project, referencing "traditional" frameworks means rigid, highly prescriptive systems engineering frameworks that include formalized governance to ensure compliance. The utility of the OpenSE framework seems promising for other research experiments that require complex instrumentation and systems, but due to its novelty, has not been widely applied. As a result, the framework's efficacy is not well understood.

1.2 Problem Statement

CERN’s OpenSE framework [2] was developed to address the challenges and gaps presented by traditional systems engineering frameworks by offering a tailored alternative that aligns with the constraints of both safety and agility for research infrastructure projects. It explicitly accommodates distributed teams, resource constraints, evolving designs, and the limited engineering documentation culture typical of research and collaboration teams. Despite its conceptual advantages, the efficacy of OpenSE, particularly when compared to minimal (i.e., ad hoc or extremely lightweight and unstructured) or traditional frameworks in scientific infrastructure projects outside of accelerator environments, remains largely unstudied. As of this initial dive into the literature, there are a few empirical studies on accelerator facility hosted research infrastructure exist. However, there is a lack of comparative, simulation-based studies assessing the impact of different systems engineering frameworks, such as OpenSE or NASA SE, on project outcomes in research infrastructure contexts.

1.3 Purpose and Research Questions

1.3.1 Purpose

The purpose of this project is to understand how framework selection affects project performance on research infrastructure projects hosted within safety-critical facilities, narrowing focus to investigate projects where technical complexity and interdisciplinary coordination are increasing but have not yet tipped the scale into large infrastructure or mission-based project space, those called medium-scale research infrastructure projects. The project aims to compare the OpenSE framework with both traditional and minimal systems engineering approaches to determine their relative impact on project performance and outcomes. By constructing a virtual project model based on representative experimental physics project data, the project will simulate and compare project outcomes under three distinct framework conditions:

1. a minimal workflow reflecting ad hoc practices,
2. the OpenSE framework, and

3. a traditional systems engineering framework, as defined by the NASA Systems Engineering Handbook.

To evaluate the practical value of OpenSE in this type of environment, a representative project that captures these challenges is necessary for empirical model and simulation inputs and evaluation against actual project performance for follow on work to this project. The COHERENT collaboration at the Spallation Neutron Source (SNS) at Oak Ridge National Laboratory (ORNL) provides an ideal representative case of this class of medium-scale research infrastructure projects that integrate into a facilities that produce ionizing radiation. COHERENT integrates multiple detector technologies and multi-institutional teams to study coherent elastic neutrino–nucleus scattering (CEvNS) [6], requiring coordination with both the SNS facility and across other COHERENT detectors. Projects of this nature demand a framework that will balance flexibility for scientific discovery with the engineering discipline necessary to deliver reliable, safe, and verifiable systems.

1.3.2 Project Scope

This master’s project is limited to an initial literature study, a crude model and simulation of a virtual project, and an initial analysis effort. An end goal of this project is to effectively set up the foundation for follow-on work to include subsequent research with a deeper literature review, higher fidelity simulation and evaluation of model predictions against real project performance.

The specific objectives of this master’s project are to:

- Conduct a literature review of current OpenSE applications in scientific infrastructure projects.
- Analyze and characterize structural differences between minimal, OpenSE, and NASA SE frameworks.
- Develop and execute a virtual project simulation using representative data from detectors belonging to the COHERENT Collaboration.
- Perform a comparative analysis of framework impacts on simulated project outcomes.

1.3.3 Project Questions

Primary Question

How do the OpenSE, NASA Systems Engineering Handbook, and minimal workflow frameworks influence simulated project success metrics in medium-scale research infrastructure and experimental physics projects?

Supporting questions:

1. What qualitative and/or quantitative evidence exists to demonstrate impact of systems engineering as a practice on medium-scale research infrastructure or equivalent projects?
2. What are metrics of project success for medium-scale research infrastructure projects?
3. What elements of a project are well-understood to impact success metrics?
4. What are the characteristics of each framework that map to project elements that impact success?

2 Literature Review

This initial literature review supports this project by identifying current practices and characteristics of the frameworks in question, providing foundational metrics to build a virtual project, and synthesize the current understanding of the application of systems engineering frameworks to R&D and scientific infrastructure projects. This is a *non-exhaustive* search and review of publications and research from the last decade. It focuses on determining how different frameworks influence project structure, maturity, and performance, particularly in medium-scale research space.

The review aims to just begin to answer the following guiding questions:

- What are the defining characteristics of the OpenSE framework, and how do these compare to minimal and NASA frameworks?
- What advantages or limitations have been reported in applying traditional systems engineering frameworks to scientific infrastructure projects?
- How has systems engineering been adapted for use in research-oriented scientific collaborations?
- How are research infrastructure project measuring their success and which metrics are most transferable to a simulation-based evaluation?
- Where do gaps exist in modeling and simulation-based comparisons of framework performance in scientific research contexts?

2.1 Framework Characteristics

2.1.1 NASA Systems Engineering Framework

The NASA Systems Engineering (SE) framework, formalized in NASA Procedural Requirements (NPR) 7123.1D [1] and detailed in the NASA Systems Engineering Handbook [7], defines a structured systems engineering approach that will "enhance NASA's core engineering capabilities while improving safety, mission success, and affordability." [1]. It is a multidisciplinary, lifecycle approach that integrates hardware, software, and human elements at all levels of hierarchy of a

system. The framework divides the project lifecycle into Formulation and Implementation phases, each containing well-defined reviews with specified entrance and exit criteria. Each gate verifies that a project is in compliance with NPR 7123.1D and in turn validates its intent to ensure the end product meets the customer's need [1].

The NASA SE Framework is characterized by Common Technical Processes, Tools and Methods, and Workforce. Figure 2.1 illustrates the view that the systems engineering capability increases along the increase of those 3 axes.

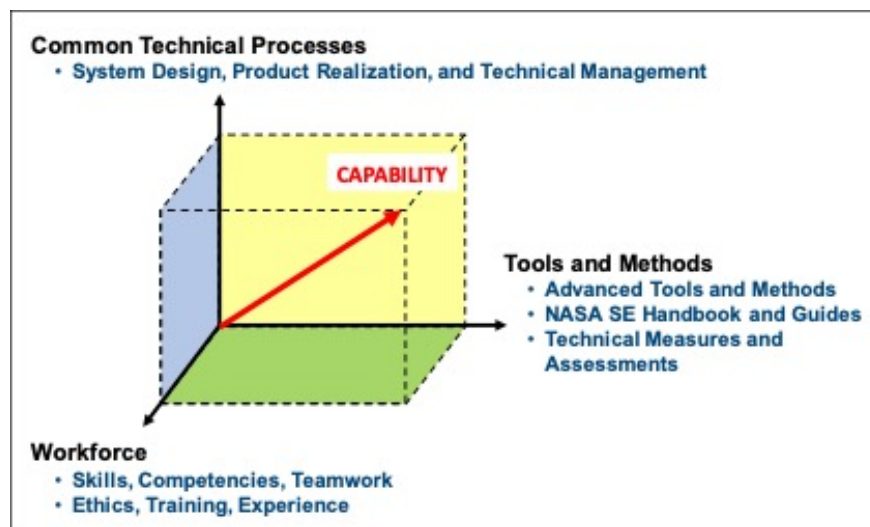


Figure 2.1: NASA Systems Engineering Framework [1].

This indicates a deeply embedded systems engineering culture that has explicit roles and job functions of systems engineers to support this framework. The framework's strengths include clear accountability, comprehensive documentation and guidance, and alignment with agency-wide standards. The common technical processes mentioned here are defined at the Systems Engineering Engine [1] that are comprise of 17 Technical Processes that are guided by the methods specified in the NASA SE Handbook [7] and other guiding documents. The 17 Technical Processes are:

1. Stakeholder Expectations Definition
2. Technical Requirements Definition

3. Logical Decomposition
4. Design Solution Definition
5. Product Implementation
6. Product Integration
7. Product Verification
8. Product Validation
9. Product Transition
10. Technical Planning
11. Requirement Management
12. Interface Management
13. Technical Risk Management
14. Configuration Management
15. Technical Data Management
16. Technical Assessment
17. Decision Analysis

2.1.2 OpenSE Framework

OpenSE was developed within CERN’s Engineering Department as a systems-engineering approach explicitly tailored for scientific facilities [2, 8]. It was motivated by the recognition that research infrastructures differ fundamentally from aerospace or defense projects: they are collaborative, distributed, constrained by limited documentation cultures and evolving experimental designs, and must balance that with a focus on safety and reliability in the face of facilities with ionizing radiation. OpenSE aims to preserve the benefits of structured lifecycle management while reducing unnecessary process overhead.

The framework defines six phases (Initialize, Study, Design, Build, Commission, and Finalize) that parallel the ISO/IEC/IEEE 15288 lifecycle [9] but with greater allowance for iteration and concurrency.



Figure 2.2: OpenSE Lifecycle [2].

The framework notes in several places the inspiration from the NASA SE Handbook but that it is more specialized to research systems exposed to ionizing radiation and emphasizes tailorability from simple to more advanced approaches to the application of this framework [2]. OpenSE emphasizes four guiding principles: openness, leanness, participation, modularity, and scalability [2]. These principles shift the emphasis of governance that NASA takes through NPR 7123.1D [1] and toward a minimal oversight expect safety kind of approach. The activities OpenSE describes are:

- Needs and Requirements
- Integration
- Verification and Validation
- Solution Finding
- Safety Documentation

and are adapted from NASA's 17 Common Technical Processes.

2.1.3 Minimal Frameworks

In contrast to structured or formal frameworks, many scientific collaborations, especially medium-scale research infrastructure, operate using minimal or ad hoc project management and subsequent systems engineering structures. These approaches are characterized by limited lifecycle documentation, informal change control, if any, and reliance on experienced team leads for integration oversight rather than formal systems engineering artifacts. Most systems engineering practices are not highlighted as practices derived from this discipline but are largely viewed as compliance measure to meet facility requirements but are often considered after large portions of the research infrastructure are fabricated and assembled. For this reason, there are not technical processes to discuss except derived characteristics which include Principle Investigator (PI)-driven design and

facility requirements. These will be referenced as Solution Finding, Integration, System Safety Verification, and ES&H Documentation.

2.1.4 Comparative Summary

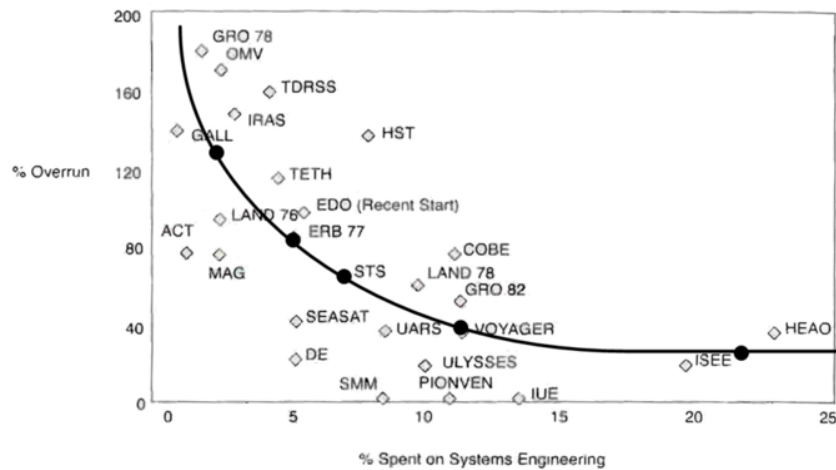
Amongst the 3 approaches, NASA SE represents the most formalized and prescriptive model, emphasizing completeness, documentation, and mission assurance. OpenSE, while rooted in the same lifecycle principles, explicitly adapts these practices for research environments, reducing overhead while preserving modularity and openness. What is not addressed in the literature or identified studies of OpenSE is the addressing the claim that OpenSE is better suited because of its tailorability. NPR 7123.3 has specific guidance on tailoring the requirements based on mission cost and complexity categorized by class (Class A–D) [1]. Further work should be done to investigate this claim. Minimal frameworks, by contrast, prioritize flexibility and speed at the cost of being formally verifiable and reproducible. The comparative characteristics of these frameworks, as synthesized from the literature, are summarized in Table 2.1.

The characteristics were selected based on common activities or functions found across each framework.

2.2 Advantages and Limitations of these Systems Engineering Frameworks

Advantages Many studies have been conducted over the last decade or more verifying the benefits of systems engineering. In an effort to not recreate many meta-analyses that already exist, this review will pull from well-known and established resources, primarily the Systems Engineering Book of Knowledge (SEBoK) [3]. In the review of the economic benefits of systems engineering in SEBoK, a study highlighting the criticality of systems engineering for NASA missions is discussed. Figure 2.3 illustrates the relationship between the percentage of project resources devoted to systems engineering and the percentage of project cost overrun [3].

Data from NASA programs show that projects investing less than 5% in systems engineering ex-



*Source: Werner M. Gruhl, Chief Cost & Economics Analysis Branch, NASA Headquarters

Figure 2.3: Relation of SE Investments to NASA Program Cost Overruns from SEBoK [3].

perienced overruns exceeding 150%, while those allocating greater than 15% maintained very low overruns. This trend demonstrates that robust systems engineering practices reduce technical and programmatic with some enhanced cost and schedule predictability. According to this study, even a modest increase in systems engineering investment yields significant benefits in project controls success. The gap here is understanding the impacts on this correlation if system performance or mission success were factored in.

It is evident that there needs to have a balance struck on time and resources spent on systems engineering and complexity of a system or project [3]. The relationship between the percentage of project time devoted to initial architectural design and risk resolution versus the total schedule impact, including rework, is illustrated in Figure 2.4.

There is clear optimum or "sweet spot" where fairly moderate early investment in architecture and risk management minimizes overall schedule growth. Projects that spend too little time in these early phases face substantial rework later, while excessive early analysis can also extend schedules unnecessarily. Physicist and engineers alike can fall prey to "analysis paralysis." Effective systems engineering finds the balance, ensuring that sufficient effort is applied early to reduce downstream uncertainty and rework, ultimately improving schedule efficiency and project assurance. Most sys-

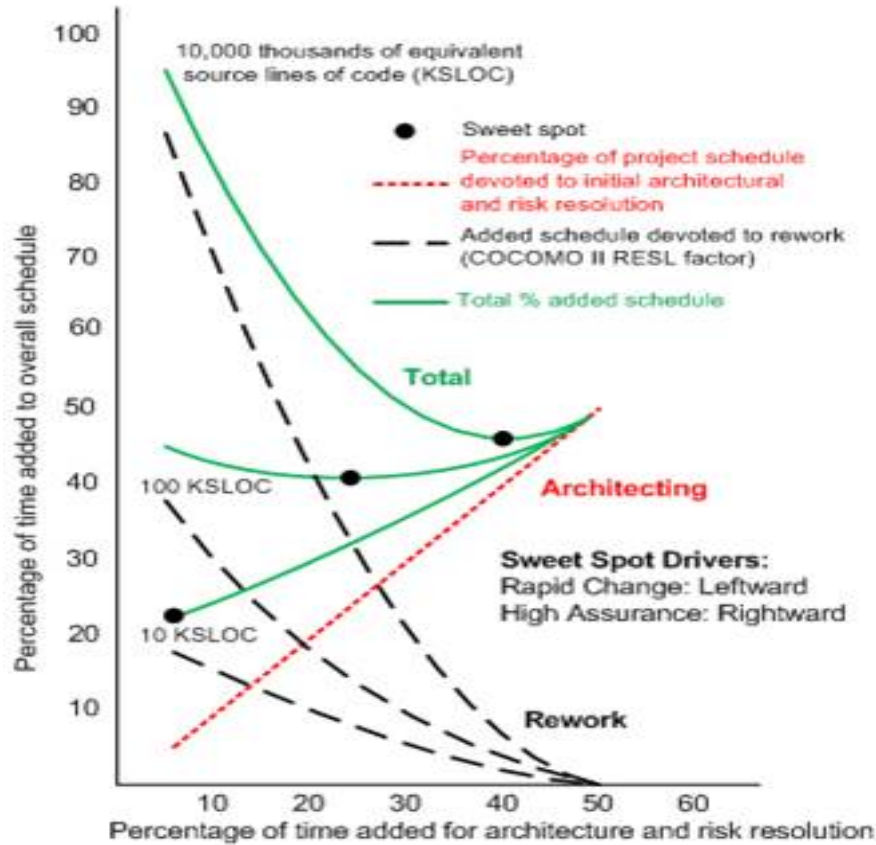


Figure 2.4: Risk-Balanced “How Much SE Is Enough” Plot from SEBoK [3].

tems engineering frameworks, if structured well, provide precisely that: tailorability in application (even the more rigid frameworks). And that is at the heart of what question being asked here - what is "enough" for medium-scale research infrastructure projects and is OpenSE a viable tailored framework? Bonnal et al [8] describes OpenSE as “particularly suited to particle-accelerator studies and development projects,” noting its flexibility for projects with shared governance and partial reuse of existing infrastructure (i.e., facility infrastructure from previous experiments).

Limitations Multiple analyses have noted that, though there are strengths of prescriptive and specific frameworks, rigid systems engineering can impose, real or perceived, unnecessary administrative overhead and reduce flexibility when applied to small or exploratory projects [5] [10] [11]. The rigor of NASA SE could exceed the practical capacity of medium-scale research infrastructure projects or research-driven teams, particularly those operating with evolving scientific objectives or limited dedicated engineering staff. Kaeske et al [12] discuss the usefulness of the framework but

highlight the needs for more prescriptive methodology and tool support to bridge the gap between intentional generalizations and practical guidance. Honoré-Livermore et al [11] and Hodges and Granados [10] highlight that early-stage R&D projects often view traditional systems engineering as incompatible with their culture of rapid iteration and discovery. Instead, these early-stage R&D projects apply systems engineering principles selectively, focusing on design and documentation only when required for compliance with a funding organization or the facility hosting the system.

Identified Gaps and Solutions for R&D There is this well-known term called the “Valley of Death” that typically arises at Technology Readiness Levels (TRLs) 5 and 6. This valley is where responsibility transitions from early-stage research to full-scale development. This phase often becomes a point of friction between exploratory R&D engineers and those focused on later development at TRLs 7 and higher [5]. Many efforts fail to progress beyond this stage due to insufficient funding, performance gaps, or inadequate mechanisms to bridge research outputs with operational capability. Additional causes include unclear research objectives (requirements), poorly managed risks, weak coordination between transitioning organizations (knowledge and interface management), and ineffective application of processes (training and methods) [5]. Misaligned performance measures, leadership conflicts, and entrenched cultural factors can further exacerbate the issue. Ultimately, the findings keep coming back to the importance and impact of selecting and implementing the right framework for the *project at hand*.

2.3 Research Infrastructure Project Metrics

So how do we choose the right framework for the *project at hand*? The first step might be to characterize what goals the project is trying to achieve and how it intends to measure its success at getting to those goals. Limited studies or interviews exist to quantify how research experiments, beyond individual experiments of physics runs, quantify success, highlighted by Stotz and Drews [13]. The metrics they did discover after their study were grouped into six categories:

1. **Volume:** scale and reach of experimentation
2. **Outcome-Based:** results and impact of experimentation

3. **Quality:** rigor and reliability of experimentation
4. **Engagement and Adoption:** how widely experimentation practices are embraced
5. **Process Efficiency:** duration from experiment setup to actionable results
6. **Strategic Alignment:** impact on broader organizational goals

These categories can easily be mapped back to typical project metrics that give a foundation for the simulation-based study. Volume translates to uptime, outcome-based translates to earned value or cost, quality translates system performance and data integrity, engagement and adoption translates to customer satisfaction, process efficiency translates to schedule, and strategic alignment translates to stakeholder satisfaction.

2.4 Gaps in Simulation-Based Research

To date, this project's efforts have not found any studies or research using simulation-based methods to compare systems engineering frameworks impact to performance in the context of scientific research infrastructure projects. Further efforts should be made to find alignment between research that exists in similar domains to inform the future work performed.

Table 2.1: A Summary of Common Elements Across Frameworks

Characteristic	NASA SE	OpenSE	Minimal
Lifecycle Structure	Sequential and gated	Iterative and facility-driven	Informal or project-defined
Problem Definition	Stable once defined. Significant resources dedicated to requirements development.	Functional requirements evolve while safety requirements are stable. Moderate resources dedicated to requirements development.	Minimal time spent on requirements development. Reactive to safety and facility requirements.
Interface Management & Integration	Formalized. Governed by ICDs.	Structured but more flexible. Focuses on collaboration and shared understanding.	Implicit and unstructured. Usually through verbal agreements or minimal documentation, as required by facility.
Verification & Validation	Formalized, following required V&V plans.	Formalized, following required V&V plans.	Informal or reactive to facility verification needs. Validation relies on in situ testing.
Flexibility	Low to moderate. Suited for stable requirements.	Moderate to high. Suited to evolving designs.	High to very high. Suited for rapid development and changes.
Type/Domain	First of a kind and high-stakes/aerospace, defense, missions.	First or one of a kind, long-term, and iterative/research infrastructures, accelerators.	First of a kind, demonstrators or prototypes/detectors, instruments, and bench-top R&D labs.
Design & Solution Finding	Formalized and driven by agreed-upon requirements. Robust, multidisciplinary engineering studies.	Somewhat formal with robust engineering. Driven by needs and requirements. Driven by physicists who act as both customer and engineer.	Informal. Driven by PI needs. Driven by physicists who act as both customer and engineer.
Documentation & Configuration Management	Extensive, standardized artifacts. Robust engineering controls.	Lean, scaled to project size. Engineering controls.	Minimal, variable by team. Some or little engineering controls.

3 Methods

Preamble and Caveat First and foremost, it is important to note that all parameter distributions are illustrative and chosen to reflect qualitative differences among frameworks rather than measured quantities. Values are based on engineering judgment and prior experience in research infrastructure projects, specifically COHERENT. The model is intended to demonstrate structure and possible sensitivity, not to forecast actual science yield. A recommended next step is to create dynamics loops that better represent behaviors, replace these assumed distributions with data from project lessons learned or published later applicable systems engineering effectiveness studies.

A Monte Carlo simulation approach was implemented to estimate the expected distribution of science data loss across three systems engineering (SE) frameworks: NASA SE, OpenSE, and a Minimal. The simulation was performed in Python using NumPy, Matplotlib, and Pandas libraries for numerical computation, visualization, and data aggregation, respectively. A total of $N = 100,000$ iterations were executed per framework to ensure statistical stability in the resulting distributions. Each iteration generated an artificial realization of eight independent parameters representing major project performance drivers, summarized in Table 3.1.

These include duration overrun (D), rework fraction (R), integration readiness level (IRL), maintenance downtime (T_m), design compromise severity (P), budget reallocation due to rework (B_1), inefficiencies and overburden (I), and measurement quality degradation (Q). Each parameter was expressed as a normalized fraction of impact (0–1) or readiness (for IRL), allowing combination into a unified performance index. These parameters are an initial derivation from the literature review.

For each framework, the parameters were sampled from either bounded normal or triangular distributions. These distribution shapes were chosen based on the level of uncertainty and skew expected from the framework’s project management characteristics and how well studied the implementation of these approaches are. For example, NASA SE emphasizes documentation rigor, leading to slower progress but lower variance in integration readiness, whereas Minimal exhibits greater

Table 3.1: Simulation Variables and Descriptions

Variable	Symbol	Description and Rationale
Duration Overrun	D	Fractional increase beyond baseline schedule. Reflects planning discipline and lifecycle control.
Rework Fraction	R	Portion of effort required for correcting or revisiting previous work. Indicates quality of initial execution and requirement clarity.
Integration Readiness	IRL	Scaled to $[0, 1]$, where 1 represents fully integrated, interface-mature components. IRLs are one a 1-9 scale that mirrors TRLs [14]. Captures coordination and pre-integration maturity.
Maintenance Down-time	T_m	Expected fractional time lost due to system unavailability. Reflects robustness, results of performing FMEAs, and having support engineering.
Design Compromise	P	Degree of functional or performance trade-offs made during development. Includes scope reduction and underperformance.
Budget Reallocation	B_1	Portion of budget diverted due to unforeseen work (e.g., rework, emergent tasks).
Inefficiencies and over-burden documentation	I	Fractional loss due to admin burden. Assumption based on procedural inefficiency estimates, slow decision-making.
Measurement Degradation	Q	Captures the loss of data quality due to instrumentation or procedural faults. Affects science yield directly.

uncertainty due to weak control and lack of formal processes. The varying standard deviations were informed by how characterized or studied these impacts are, in general, as well.

Each framework’s parameterization is summarized by the qualitative factors below:

- **NASA SE:** conservative lifecycle durations, high effort, and strong reliability and safety focus.
- **OpenSE:** iterative and lean processes with moderate formality and focus on safety.

- **Minimal:** loosely coordinated workflows with higher performance, schedule, and cost volatility, but greater flexibility.

Initial simplicity was the goal and thus the overall science data loss fraction (L) was computed as the normalized mean of the eight contributing parameters:

$$L = \frac{D + R + (1 - IRL) + T_m + P + B_1 + I + Q}{8}, \quad (3.1)$$

where L represents the effective fraction of total potential science yield lost due to engineering inefficiencies or integration challenges. All values were constrained within the $[0, 1]$ range.

A summary of simulation statistics was computed for each framework, including mean (μ_L), standard deviation (σ_L), 5th and 95th percentiles, and the probability of exceeding a 10% data loss threshold, $P(L > 0.1)$, which serves as a risk indicator for unacceptable performance degradation.

The full Python script used for this analysis is included in the GitHub repository **syse695-masters-planB** where this report is also maintained.

4 Results

Figure 4.1 shows the resulting probability density distributions for the simulated science data loss across the three frameworks.

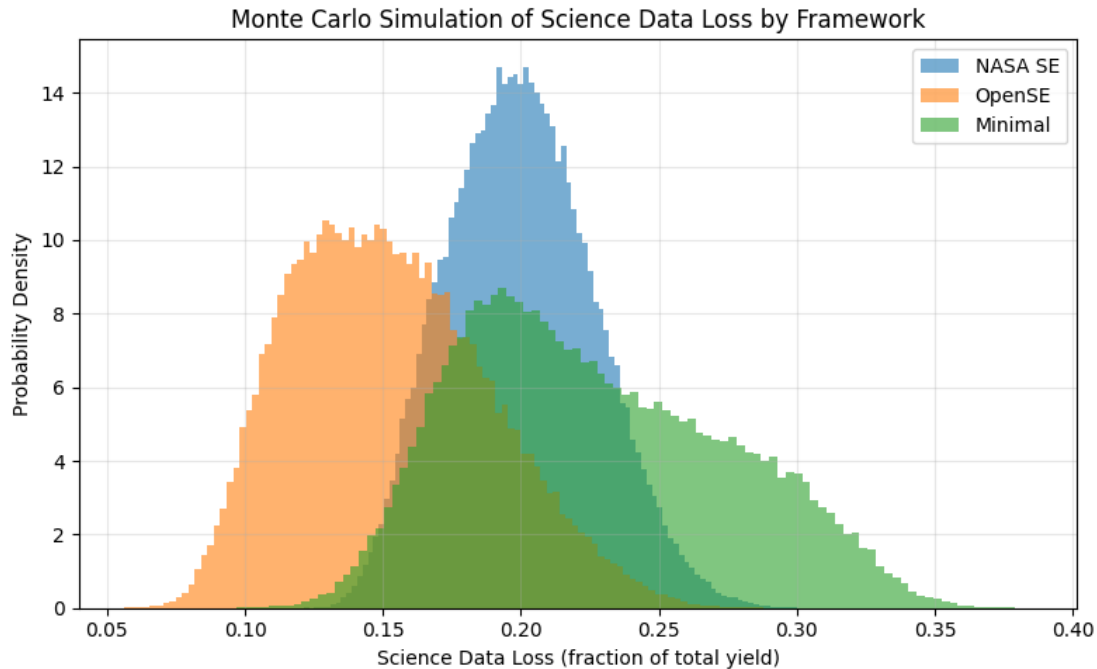


Figure 4.1: Initial Scientific Yield Loss Monte Carlo Distribution

NASA SE displays a narrow, low-loss distribution consistent with highly structured engineering control. OpenSE exhibits a broader spread with a slightly higher mean loss, reflecting its iterative and flexible implementation approach. The Minimal workflow demonstrates the widest spread and highest probability of significant loss, indicative of unmanaged variability and weak lifecycle controls.

The summary statistics for all three frameworks are shown in Table 4.1.

Table 4.1: Summary of Statistics

Metric	NASA SE	OpenSE	Minimal
Mean L	0.200	0.152	0.226
Std. L	0.026	0.036	0.048
5th Percentile	0.158	0.100	0.157
95th Percentile	0.245	0.216	0.312
$P(L > 10\%)$	1.000	0.950	1.000

The probability of exceeding 10% science data loss, $P(L > 0.1)$, was lowest for NASA SE and highest for the Minimal case, with OpenSE positioned in between—suggesting that moderate formalism combined with agility may reduce risk without incurring excessive process overhead.

5 Discussion

The results of the Monte Carlo simulation provide insight into the expected science data loss across three systems engineering frameworks: NASA SE, OpenSE, and a Minimal framework.

Overall, the OpenSE framework demonstrates the lowest mean simulated science data loss ($\mu = 0.152$), followed by NASA SE ($\mu = 0.200$), with the Minimal framework performing worst ($\mu = 0.226$). While NASA SE is traditionally associated with high procedural rigor and reliability, its high burden of procedural overhead may offset its gains in schedules where quicker turn around and flexibility are often required for exploratory research projects.

Despite OpenSE's looser structure, it benefits from lower rework levels and a balance between flexibility and formality, which likely contributes to its lower science loss. The broader variance (Std. $L = 0.036$) suggests more sensitivity to project conditions but still maintains "acceptable" performance.

In contrast, the minimal framework shows both higher mean loss and greater variance (Std. $L = 0.048$), confirming expectations about the risk of adopting ad hoc approaches for medium-scale projects where complexity is increasing. The 95th percentile loss for Minimal (0.312) indicates a long tail of worst-case scenarios, reflecting a possible increased risk in scientific output degradation.

The probability of exceeding a 10% science loss threshold is nearly certain for all frameworks ($P(L > 10\%) > 0.950$), but OpenSE is the only framework that shows any meaningful chance of staying below this threshold, indicating potential suitability in environments that demand both flexibility and scientific reliability.

These findings partially reinforce a hypothesis that neither extreme formality nor minimal structure is optimal for medium-scale research infrastructure. Frameworks like OpenSE, designed for scientific environments, may offer a better trade-off between structure and agility.

Bibliography

- [1] National Aeronautics and Space Administration. *NASA Procedural Requirements 7123.1D: NASA Systems Engineering Processes and Requirements*. Washington, D.C., July 2023.
- [2] CERN Engineering Department. *OpenSE: A Systems Engineering Framework Tailored for Scientific Facilities*. CERN, 2016. Version 1.0.
- [3] INCOSE, IEEE Computer Society, and Stevens Institute of Technology. Economic value of systems engineering. https://sebokwiki.org/wiki/Economic_Value_of_Systems_Engineering, 2025.
- [4] Joseph W. Harrison, Michael A. Lewis, Jens K. Roehrich, and Andrew Davies. Scale in project management: A review and research agenda. *Project Management Journal*, 55(6):708–722, 2024.
- [5] Michael DiMario, Gary Mastin, Heidi Hahn, Ann Hodges, and Nick Lombardo. Perceived conflicts of systems engineering in early-stage research and development. *INSIGHT*, 24(3):8–14, 2021.
- [6] Jason Newby. What is coherent elastic neutrino-nucleus scattering? <https://www.ornl.gov/division/pd/coherent>, 2025. Physics Division, Oak Ridge National Laboratory.
- [7] National Aeronautics and Space Administration. *NASA Systems Engineering Handbook*. NASA/SP-2016-6105 Rev 2. NASA Headquarters, Washington, D.C., 2016. NASA/SP-2016-6105 Rev 2.

- [8] P. Bonnal, B. Féral, K. Kershaw, B. Nicquevert, M. Baudin, L. Lari, and J. Le Cardinal. opense: A systems engineering framework particularly suited to particle accelerator studies and development projects. *arXiv preprint arXiv:1801.03836*, 2018.
- [9] Iso/iec/ieee 15288:2023 systems and software engineering — system life cycle processes, 2023.
- [10] Ann Hodges and Arno Granados. A bridge blueprint to span the chasm between research and engineering — a framework for systems engineering in early-stage research and development. *INSIGHT*, 26(3):15–25, 2023.
- [11] Evelyn Honoré-Livermore, Knut Robert Fossum, and Erik Veitch. Academics’ perception of systems engineering and applied research projects. *Systems Engineering*, 25(1):19–34, 2022.
- [12] Jens Kaeske, Erik Wagner, Albert Albers, and Stephan Russenschuck. Tool support for implementing a methodology in magnet development projects at cern. *Proceedings of the Design Society*, 4:2615–2624, 2024.
- [13] Nils Stotz and Paul Drews. Metrics for experimentation programs: Categories, benefits and challenges. In Sibylle Peter, Martin Kropp, Ademar Aguiar, Craig Anslow, Maria Ilaria Lunesu, and Andrea Pinna, editors, *Agile Processes in Software Engineering and Extreme Programming*, pages 210–225, Cham, 2025. Springer Nature Switzerland.
- [14] U.S. Government Accountability Office. Technology readiness assessment guide: Best practices for evaluating the readiness of technology for use in acquisition programs and projects. Technical Report GAO-20-48G, U.S. Government Accountability Office, January 2020.