Maiden in the Tower: The Intersection of MBSE and Cost Estimating (TCI05)

Stephen Koellner, Daniel Larison

2025-02-17

Abstract:

Model based systems engineering (MBSE), and more broadly digital engineering, is becoming a more widely adopted practice within federal/defense acquisition. The motivations behind this adoption are to produce more effectively engineered systems and reduce the likelihood of costly rework in the later stages of system design and integration. Cost estimators and other project control professionals are most effective when fully integrated within the overarching Systems Engineering & Program Management (SEPM) team. What do MBSE and project control practitioners have in common? The answer is that both are effectively "jack-of-all trades" with significant overlap in goals and processes. An unfortunate reality is that neither is regularly well-integrated with the other, despite the many philosophical and procedural commonalities. Through this topic, a sample case study will be presented to illustrate how these skillsets can operate in conjunction to produce more accurate and credible cost estimates as well as more dynamic and rapid evaluation of architecture/design alternatives.

Contents

List of Figures

List of Tables

Introduction

In 1812, the German academics, known as The Brothers Grimm, published a series of collected folktales such as Cinderella, Hansel and Gretel, Rumpelstiltskin, Sleeping Beauty, and Little Red Riding Hood. Another notable piece of folklore in this collection is "Rapunzel", where the titular character is locked in a desolate tower by an evil sorceress until a prince discovers her while riding through the woods. Through a complicated and occasionally dark series of events, the two eventually escape the clutches of the sorceress and live happily ever after. This story is best known for its iconic line that first enables the prince to climb up the tower: "Rapunzel, Rapunzel, let down your hair."

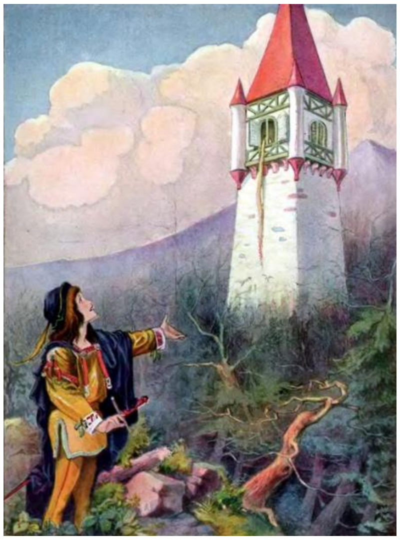


Figure 1: Depiction of Rapunzel by Johnny Gruelle

Cost estimators and systems engineers would benefit from a "let down your hair" moment. Those who work in cost estimation and scheduling may find thematic parallels between this story and the act of obtaining meaningful system data for their cost models. Systems engineers may find the incorporation of cost analysis in design decision making to be a bottleneck. It is not uncommon for cost estimators and systems engineering teams to operate mostly isolated from one another, with information transfer being an arduous and manual process rather than one which utilizes increased communication and linkages between digital models. As Digital Engineering (DE) practices become more widely adopted across capabilities acquisition, it is pivotal for there to be a "let down your hair" exchange between these two fields.

Model-Based Systems Engineering (MBSE)

DE is an overarching approach to capability acquisition that "uses authoritative sources of systems' data and models as a continuum across disciplines to support life cycle activities from concept through disposal." [1] Fields such as modeling & simulation, model-based system engineering (MBSE), model-based testing, and other related disciplines fall under the DE umbrella. MBSE specifically is "an approach to engineering that uses models as an integral part of the technical baseline that includes the requirements, analysis, design, implementation, and verification of a capability, system, and/or product throughout the acquisition life cycle." [2] A simpler alternative definition is that MBSE is an approach to systems engineering that relies on the use of an authoritative system model.

To truly understand MBSE, one must have a solid understanding of the underlying systems engineering ideology and heuristics. There are multiple views of systems engineering, such as the V Diagram, but Figure 2 represents a generalized approach to systems engineering that relies on iterating through definition/development phases while maintaining traceability between all aspects of system description such as form, function, requirements, and operational performers to the root cause operational need.

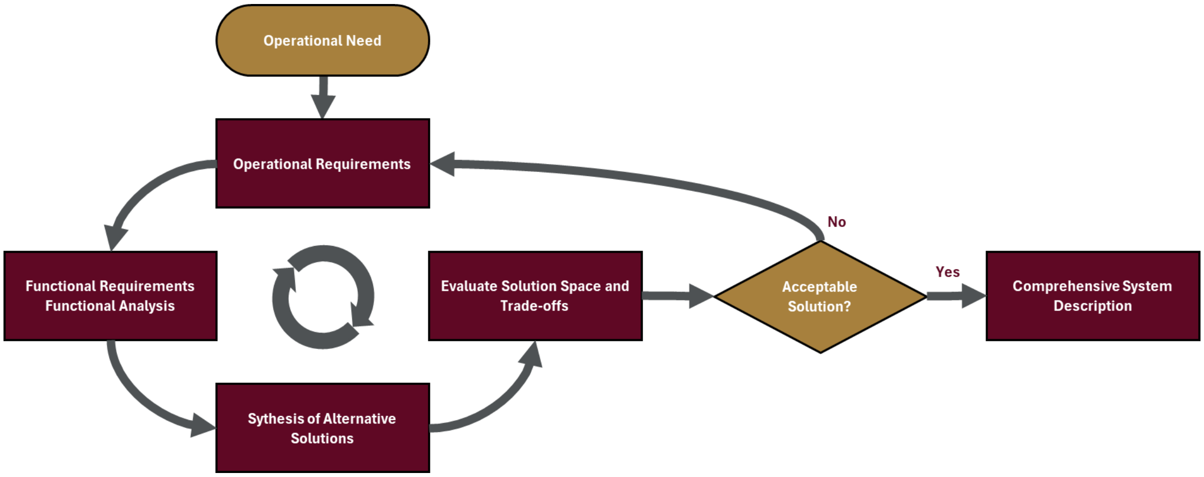


Figure 2: Generalized Systems Engineering Process

MBSE implements these overarching systems engineering processes by leveraging an authoritative single source of truth system model. The system model is a relational database that can define traceability between reusable model assets as well as representing different viewpoints of the system. These viewpoints aid stakeholders and decision makers in refining the system requirements and description throughout all stages of the system lifecycle. Architecture definition, system definition, requirements, lifecycle support, system production, and program management can all be contained within the system model (as illustrated in Figure 3) with stakeholder input and other program resources serving as the basis for these definitions.

The overarching, motivating factor for implementing MBSE within Department of Defense (DoD) acquisitions is to reduce lifecycle cost by decreasing the amount and likelihood of engineering rework in the later stages of system design [3]. It is generally more cost effective to invest in MBSE early in project lifecycle, as it is inexpensive to redesign system elements in a digital modeling environment rather than with a physical prototype/early production unit.

Figure 3: MBSE System Model

Systems Engineering V-Model

Practitioners of Systems Engineering and MBSE are familiar with the V-Model: a depiction of system life from early identification of need through development and into operations and maintenance. Figure 4 is a diagram of the V-Model, which shows this transformation throughout lifecycle. The horizontal axis shows development through time, and the vertical axis represents the level of detail

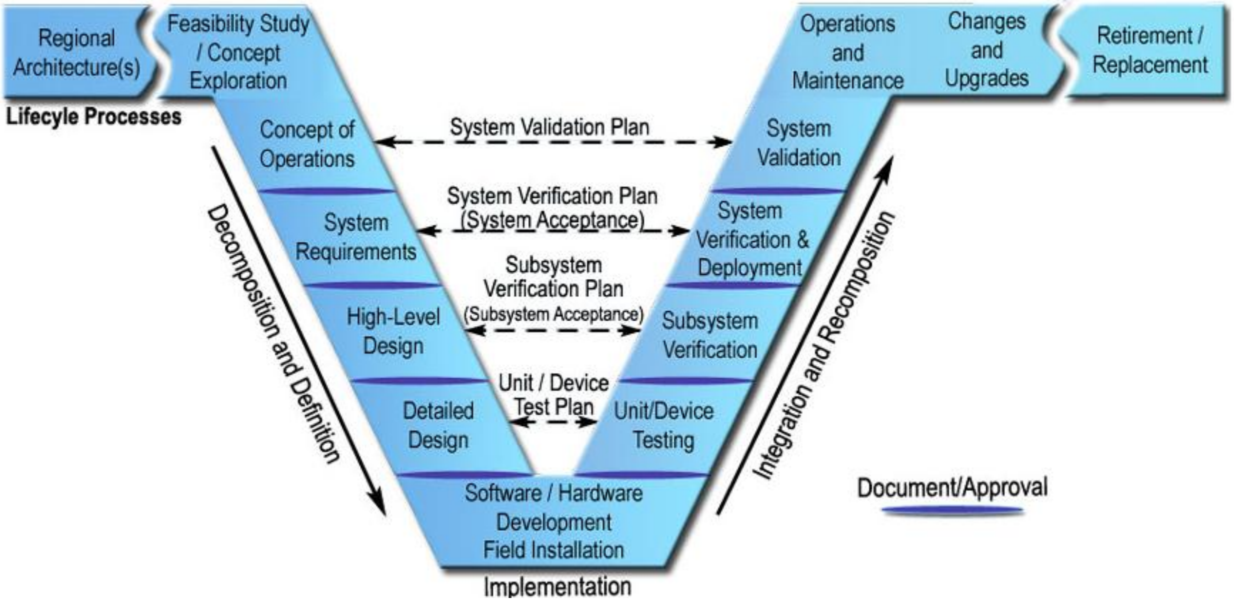


Figure 4: Systems Engineering V-Model Diagram

necessary to perform certain actions through development (high-level view at the top and detailed view at the bottom), with a focus on backwards validation & verification [4].

The V-Model begins with a system-level definition of the problem and how the system will solve this problem. The overarching functions and operational usage of the system are defined first, and then additional levels of detail are defined. System-level functions lead to necessary subsystems to support these system-level functions, subsystem functions lead to necessary subassemblies to support subsystem functions, etc., until the system is defined to the lowest possible set of components. Using this detailed description of the system, integration and testing efforts verify and validate that the designed system performs as intended, and that the system addresses the initial needs of the stakeholders.

Motivations to Integrate MBSE & Cost Estimation

The Government Accountability Office (GAO) defines cost analysis as “the systematic and rigorous application of cost analysis methods [that] provides critical support to program managers and decision authorities.” [5] Cost estimating is a subordinate task group of cost analysis focused on developing a numerical forecast of the monetary expense for a project. The GAO Cost Estimating and Assessment Guidebook outlines a twelve-step cost estimating process to produce a comprehensive, well-documented, accurate, and credible cost estimate.

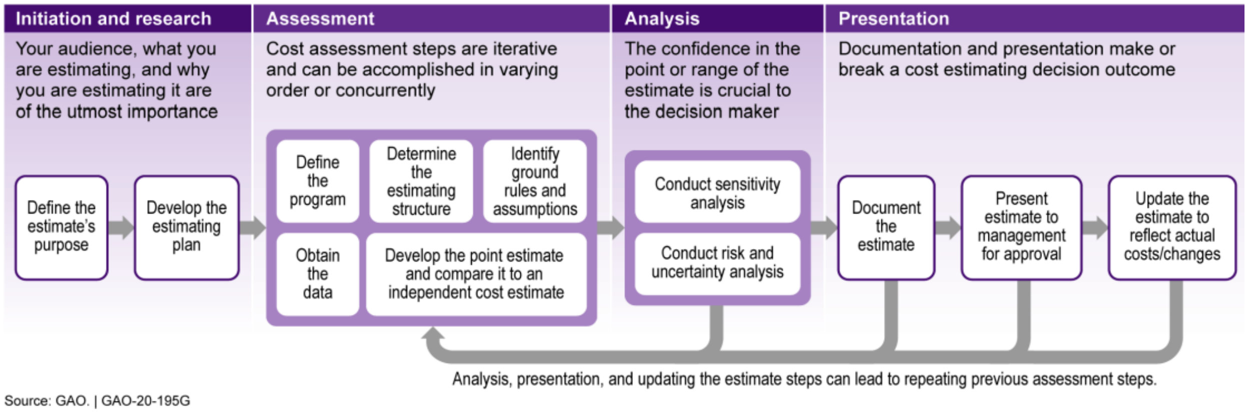


Figure 5: GAO Cost Estimating Process

Figure 5 outlines these twelve steps across four primary and iterative phases:

1. Initiation and research: bound the scope of the estimate and identify the goal of developing the estimate.
2. Assessment: a highly iterative series of steps that defines the program from a top-level assessment, and drills down into lower level definition.
3. Analysis: risk, uncertainty, and sensitivity analysis are performed to establish a confidence range around the point estimate and identify underlying driving variables in the model.
4. Presentation: formalize the estimate, methodology, and findings in an appropriate delivery artifact. This is followed by iterative updates to account for actual data and shifting strategy changes.

It is simple and intuitive to draw parallels between the V-Model (Figure 4) and standard lifecycle cost estimating practices, through both individual application and industry-wide perspectives. Individual lifecycle cost estimates identify the needs of the estimate, as well as the needs of the system being estimated. Early lifecycle cost estimates are typically informed using high-level views of the system and leverage analogies and parametric methodologies. The level of cost estimate uncertainty should also be high to account for potential variabilities in the future detailed system definition. Throughout the lifecycle of the cost estimate, an increased level of detail is utilized to inform the estimate, in alignment with the system definition, and more precise methodologies with lower variability are employed. Cost estimates maintain traceability to older versions of the estimate, and its stated purpose needs to be continuously fulfilled while being developed.

With these process parallels between MBSE and cost estimation identified, and high potential for integration, it is necessary to outline motivations and goals of potential integration. Table 1 lists and defines three motivating factors of this integration.

Motivating Factor	Description
Traceability	A strength of system modeling is the ability to trace relationships between model elements such as requirements, functions, operational usage, system decomposition, etc. Traceability is also a characteristic of a reliable cost estimate. It serves to benefit both modeling disciplines by incorporating traceability between each.
Communication	Communication between management and system designers is a stated benefit of systems engineering, and cost estimators rely on communication to obtain necessary data for their models and feedback from relevant stakeholders. Integration of MBSE and cost estimation should leverage increased communication abilities through cost practitioner access to the system model.
Automation	Exporting data reports from the system model for use in the cost model or direct software linkages between these models will reduce the time it takes to update cost estimating artifacts and decrease the likelihood that human error will cause errors in the cost model.

Table 1: MBSE & Cost Estimation Integration Motivating Factors

Literature Review

To demonstrate the disconnect between the cost estimating and digital engineering communities, a literature analysis and review was performed on existing ICEAA papers and presentations [6]. The International Cost Estimating & Analysis Association (ICEAA) archives has 1,399 total topics between 2007 and 2024, of which only 6 topics reference the tags "Digital Engineering" or "MBSE" in their title or abstract. Similarly, the International Council on Systems Engineering (INCOSE) library has 4,965 total topics between 2010 and 2024 [7]. Only 25 of these INCOSE topics include the keyword "cost estimating." In total, there are only 0.49% of papers and presentations across both organizations that overlap in these search words. Figure 6 and Figure 7 illustrate this disconnect.

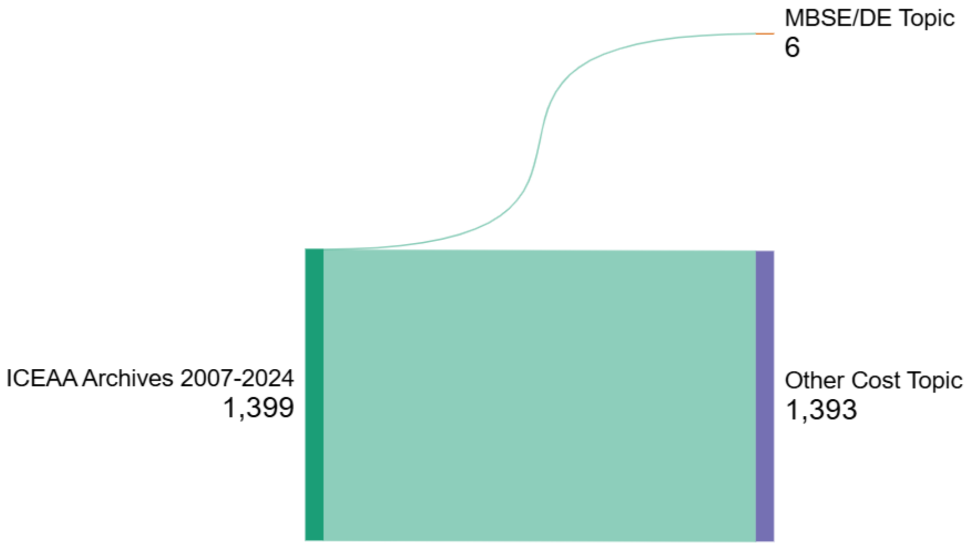


Figure 6: Sankey Chart on MBSE/DE Topic in ICEAA Papers

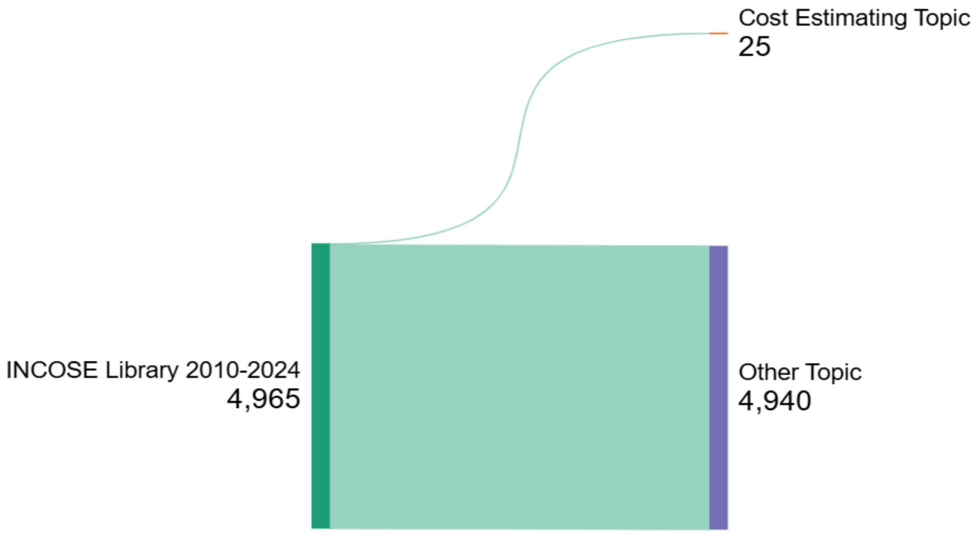


Figure 7: Sankey Chart on Cost Estimating Topics in INCOSE Papers

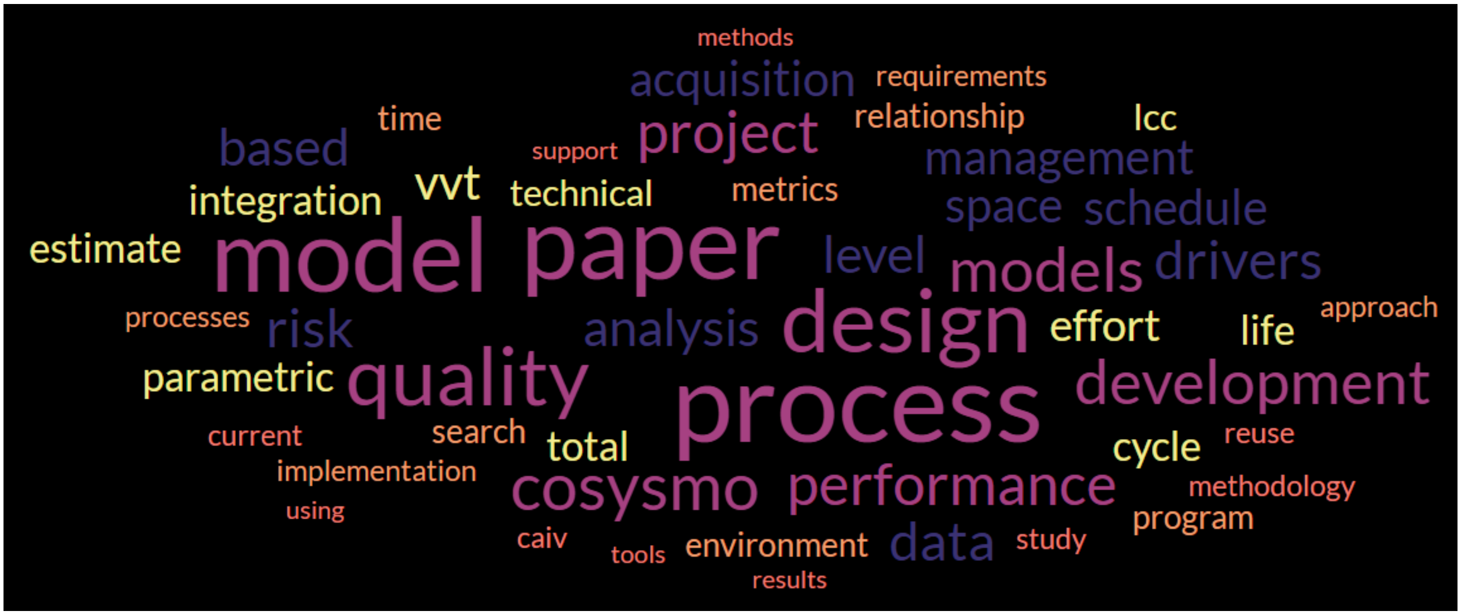


Figure 8: Word Cloud of Overlapping ICEAA & INCOSE Abstracts

A word cloud was generated from the available ICEAA and INCOSE abstracts, with search phrases (“cost”, “estimating”, “digital”, “engineering”, and MBSE”) removed, to identify common words across relevant publications from each organization (Figure 8). The most common notable word across these abstracts is the word **process**, which is defined as “a series of actions or steps taken in order to achieve a result” in the Cambridge Dictionary [8]. This indicates that the concept of defining processes is one of significant overlap between both communities of practice.

Within the six MBSE and DE topics contained within the ICEAA archives, only two are focused on defining interfaces between both sets of teams. The goal of this paper is to outline a set of heuristics and demonstrate a potential process for integrating MBSE and cost models, agnostic of specific modeling tools.

Process Suggestions

The goal within MBSE is to implement systems engineering practices through a digital model that acts as the authoritative source of truth (ASoT) of the system throughout the capability acquisition process. For the system to be considered an ASoT for project control objectives, a process flow must be defined to transfer system model data into the cost, schedule, and risk model.

An intuitive interface medium between the two sets of models is the Cost Analysis Requirements Document (CARD), which “is a description of the salient features of the acquisition program and of the system itself.” [9] A CARD specifies, to the appropriate level (based on current system definition), information such as:

- System overview
- Interfaces with other systems
- System performance parameters and characteristics
- Program milestone schedule
- Acquisition strategy

- Time-phased quantity requirements
- Design description
- Critical technologies
- Program risks
- Intelligence mission data
- Test and evaluation strategy
- Facilities and tooling requirements
- Environment, safety, and occupational health (ESOH) considerations

Common MBSE tools (such as Innoslate, GENESYS, and CAMEO) store and manage these pieces of information within the system model and export that data into a report or series of reports. While these tools usually include functionality to prepare schedules and cost estimates, they generally do not offer the same functionality as tools specifically designed for cost and schedule analysis. Advanced practitioners may also be able to directly link the system model to the schedule and cost models in certain instances, although this is not a widely available feature. Figure 9 illustrates a practical and minimal-effort process flow to interface the system model and the project controls models.

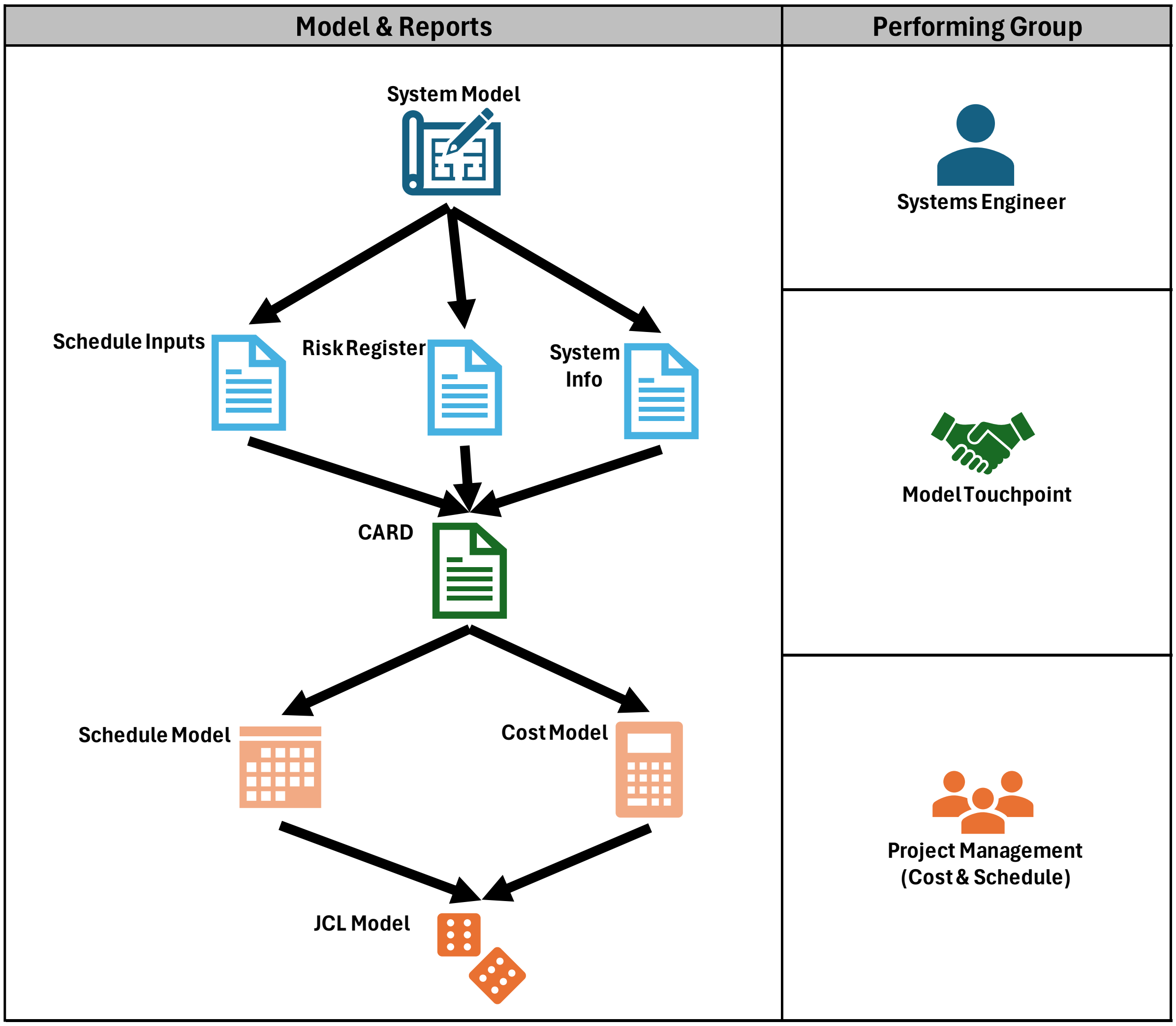


Figure 9: System Model and Project Control Model Process Flow

Cost and schedule practitioners who can directly link the system model to the schedule and cost estimate would follow a slightly different process flow (Figure 10) where the interface between these models is direct software linkage. This comes with several advantages to the approach shown in Figure 9: minimization of tasks related to manual updates, increased communication with the program systems engineering team, and faster turnaround times to update project controls model outputs. There are, however, a few considerations to keep in mind prior to the development of developing a truly integrated system & project controls model. The considerations may include restricting unauthorized automatic updates of cost/schedule outputs, the level of strategic coordination between both teams, and long-term maintenance of both model sets as system definition increases in detail. Organizations should decide in advance which process flow is most suitability for the program.

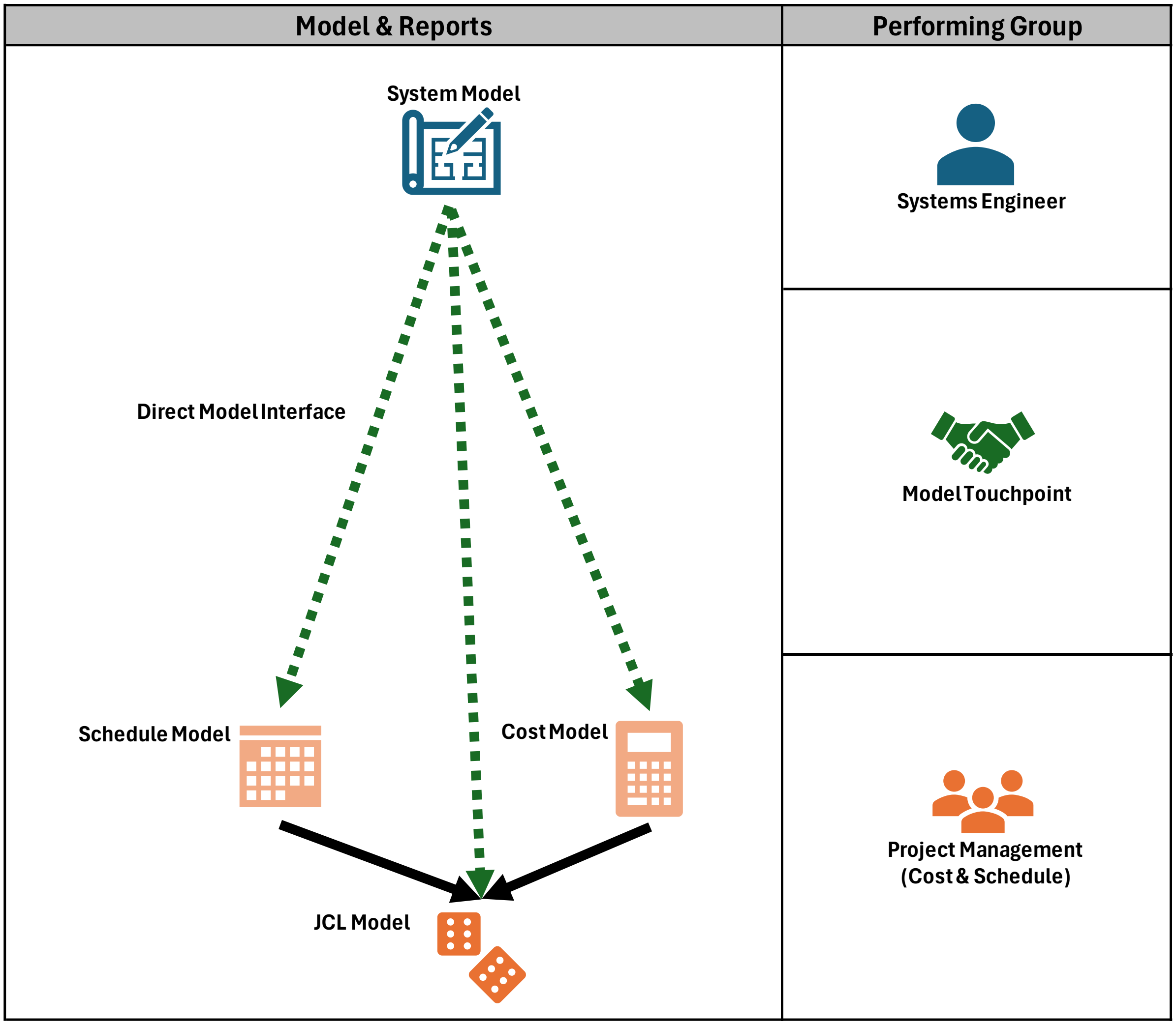


Figure 10: Direct Linkage System Model and Project Control Model Process Flow

Sample Case Study

To demonstrate the potential that integrating a system model and cost estimating artifacts has, a sample case study has been prepared using a fictional space-system program. A system model has been developed to track the requirements of the system, define operational usage of the system, decompose the system into subsystem/subassembly builds, identify program risks, and define characteristics and measures of the system relevant to the cost estimate. A cost, schedule, and Joint Confidence Level Analysis model was developed in conjunction with the system model, with built-in functionality to import relevant reports from the system model. The process flow reflects the diagram shown in Figure 9.

IRIS Program

The International Recycling In-Orbit System (IRIS) is an innovative program designed to address the growing problem of space debris. As the number of satellites and space missions increases, so does the accumulation of debris in Earth's orbit, posing significant risks to active satellites, space stations, and future missions. IRIS aims to mitigate these risks by implementing a sustainable and efficient solution for debris collection and recycling.

The IRIS spacecraft, after being launched into orbit, will navigate to the identified space debris and secure it to the spacecraft. Then the spacecraft will enter retrograde descent with the secured debris payload towards a targeted recovery point. From there, a recovery team will navigate to the collected debris to begin recycling and reuse efforts. This operation will remove debris from orbit and enable the potential reuse of non-compromised components from both the recovered debris and the IRIS spacecraft.

Pre-Planning the Integrated Environment

To begin this effort, some pre-planning is required to guide the development of both model sets. The system model and the cost model will both be initially established simultaneously, with iterative touchpoints between each to ensure synergies. The system/program structure will aid in shaping the scope of the estimate and the work breakdown structure (WBS) of the estimate. Based on this description of the system, cost analysts will determine the appropriate methodologies to estimate cost. In this instance, parametric cost estimating relationships (CERs) will be utilized to develop the cost estimate. The required input data for these CERs will be requested as exportable data from the system model.

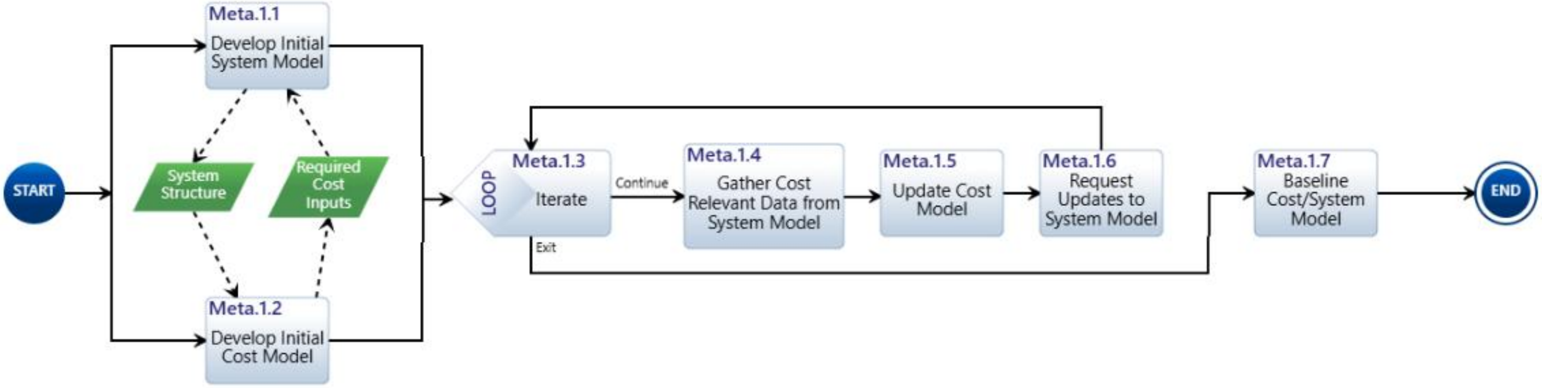


Figure 11: Integrated Environment Pre-Planning Process

An iterative process will then begin once an initial system model and cost model have been developed. Cost-relevant system data will be exported from the system model, imported into the cost model, and further refinement of both models will be identified. This iterative process will continue until a baseline cost model and system model can be established. As program life continues, both living models will continue to be iterated upon and updated.

Creating System Model

A system model for IRIS was created in Innoslate, first starting with importing overarching program requirements and a breakdown of the intended operations of IRIS. These requirements are traceable back to the needs of the stakeholders and ultimately form a hierarchy of requirements. Table 2 lists out these requirements and Figure 12 depicts these requirements in a hierarchy.

Number	Name	Description
Req	IRIS Goal	IRIS will utilize heat shield capsules equipped with advanced technology to capture, contain, and safely return space debris to Earth. The system will include ground-based facilities for processing and recycling the collected materials.
Req.1	IRIS Functional Requirements	
Req.1.1	Debris Detection and Collection	
Req.1.1.1	Debris Detection	The system shall be capable of detecting space debris of various sizes, from small fragments to larger objects.
Req.1.1.2	Collection Mechanism	The system shall include robotic arms or similar mechanisms to capture and secure debris.
Req.1.1.3	Collection Automation	The system shall be able to operate autonomously or be remotely controlled by ground operators.
Req.1.2	Capsule Design	
Req.1.2.1	Heat Shield	Capsules shall be equipped with heat shields to protect against re-entry temperatures.
Req.1.2.2	Module Capsule	Capsules shall have a modular design to allow for scalability and adaptability to different mission requirements.
Req.1.2.3	Thermal Sensors	Capsules shall include sensors and telemetry systems to monitor and report status during the mission.
Req.1.3	Re-entry and Recovery	
Req.1.3.1	Safe Re-Entry	Capsules shall be designed to ensure safe re-entry and landing in designated recovery zones.
Req.1.3.2	Precision Landing	The system shall include mechanisms for precise landing to facilitate easy recovery.
Req.1.3.3	Safe Handling	The system shall include protocols for the safe handling and disposal of hazardous materials.
Req.1.4	Ground Processing & Recycling	
Req.1.4.1	Handling	Ground facilities shall be equipped to handle and process various types of materials collected from space debris.
Req.1.4.2	Environmental Compliance	The recycling process shall comply with environmental regulations and standards.
Req.1.4.3	Safe Handling	The system shall include protocols for the safe handling and disposal of hazardous materials.
Req.2	Performance Requirements	
Req.2.1	Efficiency	
Req.2.1.1	Targeted Debris Efficiency	The system shall be capable of collecting a minimum of 90% of targeted debris within a specified mission duration.
Req.2.1.2	Fuel Consumption	The system shall minimize fuel consumption and operational costs.
Req.2.2	Reliability	
Req.2.2.1	Success Rate	The system shall have a reliability rate of at least 95% for successful debris collection and return missions.
Req.2.2.2	Redunancy	The system shall include redundancy features to ensure mission success in case of component failure.
Req.2.3	Safety	
Req.2.3.1	Safety Standards	The system shall comply with all relevant safety standards for space operations.
Req.2.3.2	Fail-Safe Mechanisms	The system shall include fail-safe mechanisms to prevent accidental release of debris during collection and transport.
Req.3	Interface Requirements	
Req.3.1	Communication	
Req.3.1.1	Comm Security	The system shall include secure communication channels for data transmission between the capsules and ground control.
Req.3.1.2	Real-Time Monitoring	The system shall support real-time monitoring and control capabilities.
Req.3.2	Integration	
Req.3.2.1	Launch Vehicle Integration	The system shall be compatible with existing space infrastructure and launch vehicles
Req.3.2.2	Ground System Integration	The system shall include interfaces for integration with ground processing and recycling facilities.
Req.4	Environmental Requirements	
Req.4.1	Space Environment	
Req.4.1.1	Space Environment Survivability	The system shall be designed to operate in the harsh conditions of space, including extreme temperatures, radiation, and microgravity.
Req.4.2	Earth Environment	
Req.4.2.1	Ground Impact	The system shall ensure minimal environmental impact during re-entry and landing.
Req.4.2.2	Sustainability	The recycling process shall aim to reduce the carbon footprint and promote sustainability.

Table 2 IRIS Requirements:

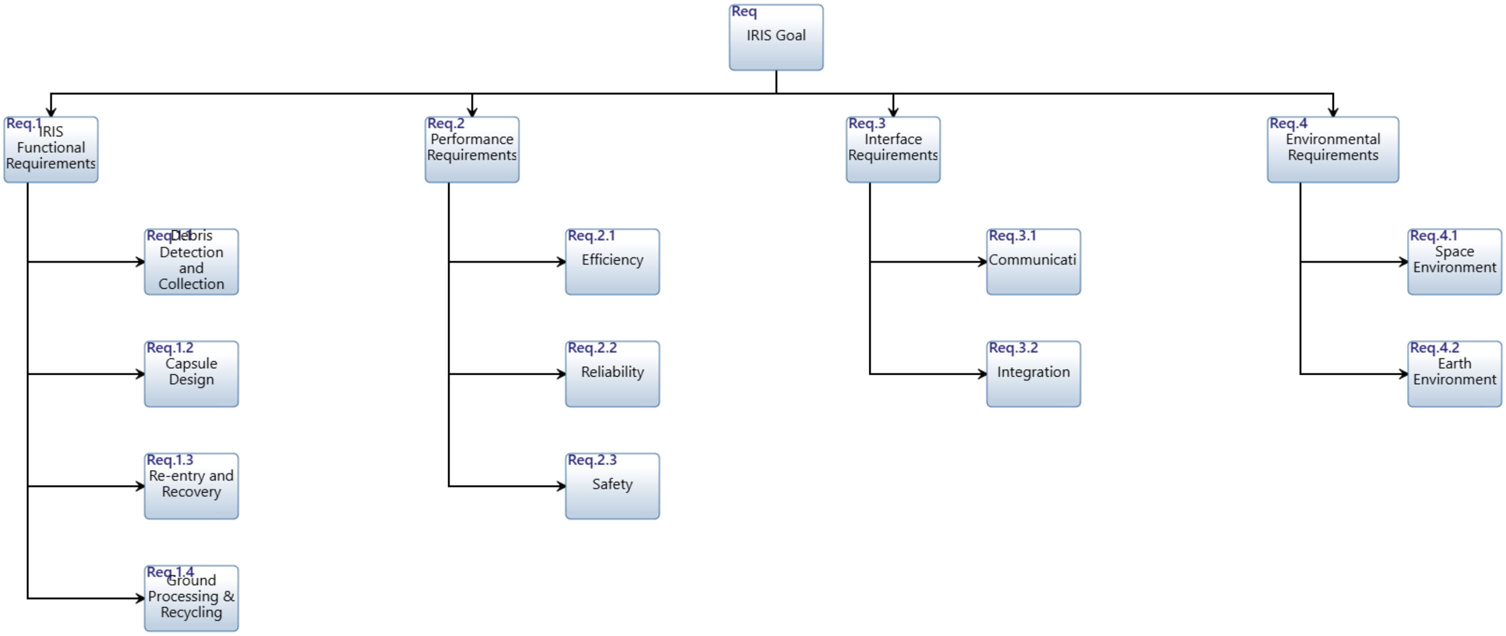


Figure 12: IRIS Requirements Hierarchy

These requirements form the basis of system functions necessary, and these functions are then organized to outline the intended system mission scenario. First, the ground control system monitors space debris until a recovery target is identified and a recovery mission is defined. Then an IRIS satellite is launched into orbit awaiting final mission initiation based on detailed sensor data from the satellite. Once the detailed mission specifications are relayed to the satellite, it will begin navigation to the debris and initiate a retrieval operation to secure the debris. With the debris secured, the satellite will enter a descent operation to a specified location without substantially destroying the recovered debris and the ground recovery team will collect and recycle components from the recovered debris and IRIS satellite. Figure 13 depicts these operations in an activity diagram captured in the system model.

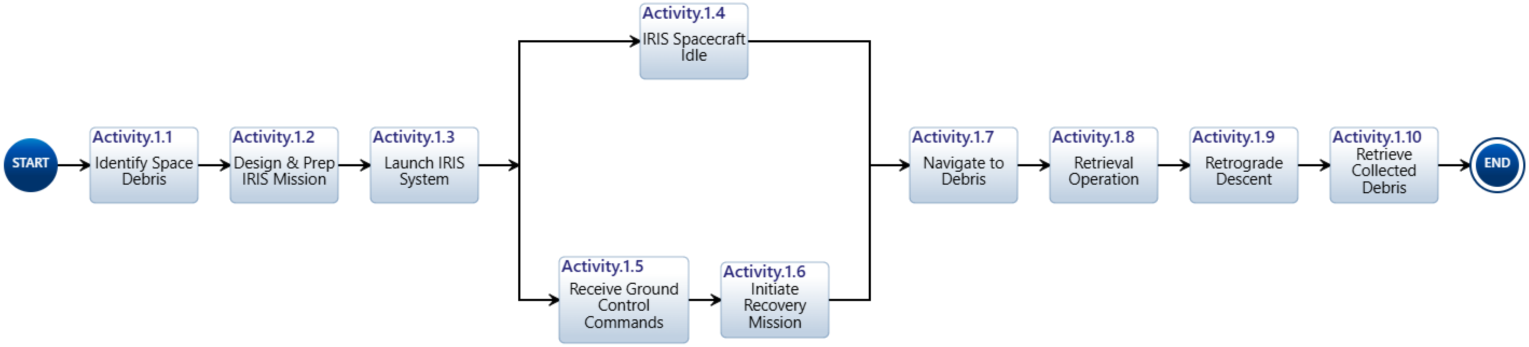


Figure 13: IRIS System Mission Operation

Mission functions of the system can be further refined into a decomposition of subsystems and components. Each subsystem and component is mappable to a function that the system must perform to fulfill mission objectives at this stage of system design. For IRIS, three primary subsystems fulfill these mission objectives: the IRIS Satellite, the Ground Control system, and the Recovery & Retrieval team. Figure 14 depicts this high level decomposition.

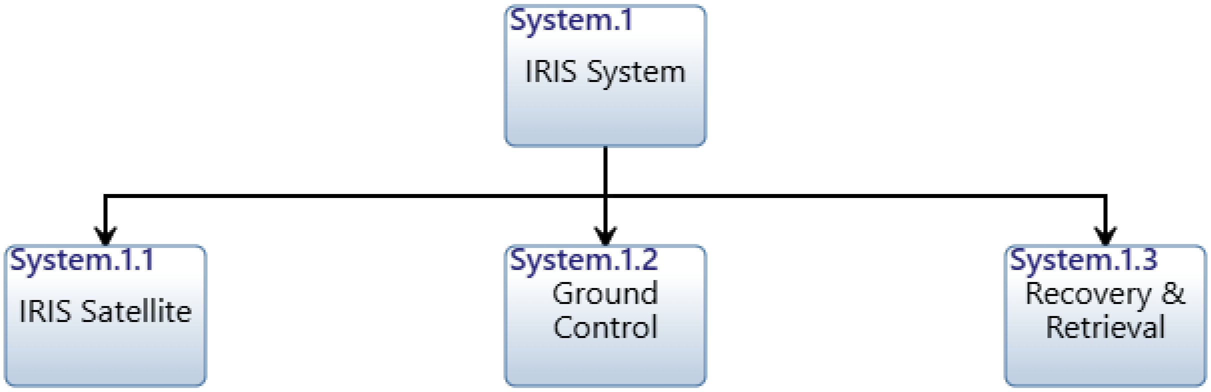


Figure 14: IRIS System Decomposition

The primary subsystem being refined in this case study is the IRIS satellite. Systems engineers will further decompose this primary subsystem, while maintaining traceability to mission obejctives.

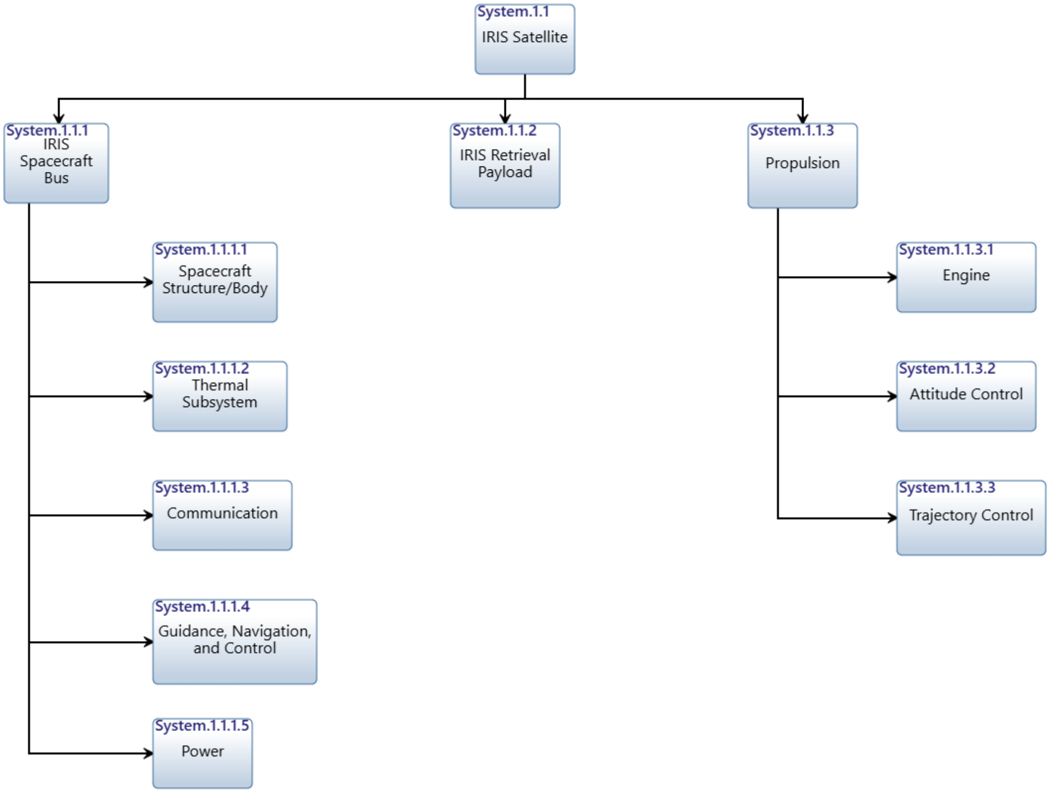


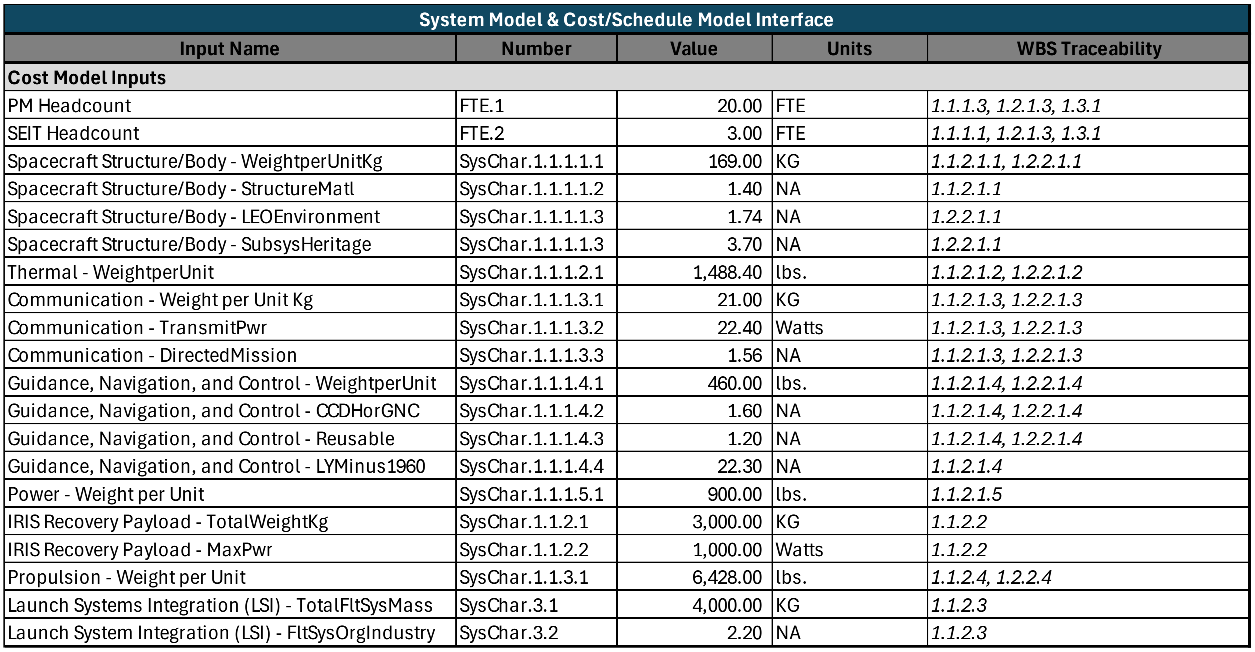
Figure 15: IRIS Satellite Decomposition

To verify traceability between system decomposition and requirements, a traceability of requirements against the system decomposition is generated from the system model (Table 3).

Requirements vs. System Hierarchy Matrix	System.1 IRIS System	System.1.1 IRIS Satellite	System.1.1.1 IRIS Spacecraft Bus	System.1.1.1.1 Spacecraft Structure/Body	System.1.1.1.2 Thermal Subsystem	System.1.1.1.3 Communication	System.1.1.1.4 Guidance, Navigation, and Control	System.1.1.1.5 Power	System.1.1.2 IRIS Retrieval Payload	System.1.1.3 Propulsion	System.1.1.3.1 Engine	System.1.1.3.2 Attitude Control	System.1.1.3.3 Trajectory Control	System.1.2 Ground Control	System.1.3 Recovery & Retrieval
Req IRIS Goal	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Req.1 IRIS Functional Requirements			X	X			X		X		X	X	X		X
Req.1.1 Debris Detection and Collection			X				X		X						
Req.1.1.1 Debris Detection			X				X								
Req.1.1.2 Collection Mechanism									X						
Req.1.1.3 Collection Automation									X						
Req.1.2 Capsule Design				X	X										
Req.1.2.1 Heat Shield					X										
Req.1.2.2 Module Capsule				X											
Req.1.2.3 Thermal Sensors					X										
Req.1.3 Re-entry and Recovery				X							X	X	X		X
Req.1.3.1 Safe Re-Entry				X											
Req.1.3.2 Precision Landing											X	X	X		
Req.1.3.3 Safe Handling															X
Req.1.4 Ground Processing & Recycling															X
Req.1.4.1 Handling															X
Req.1.4.2 Environmental Compliance															X
Req.1.4.3 Safe Handling															X
Req.2 Performance Requirements		X							X		X				
Req.2.1 Efficiency									X		X				
Req.2.1.1 Targeted Debris Efficiency									X						
Req.2.1.2 Fuel Consumption											X				
Req.2.2 Reliability		X													
Req.2.2.1 Success Rate		X													
Req.2.2.2 Redunancy		X													
Req.2.3 Safety		X													
Req.2.3.1 Safety Standards		X													
Req.2.3.2 Fail-Safe Mechanisms		X													
Req.3 Interface Requirements		X				X	X								
Req.3.1 Communication						X									
Req.3.1.1 Comm Security						X									
Req.3.1.2 Real-Time Monitoring							X								
Req.3.2 Integration		X												X	X
Req.3.2.1 Launch Vehicle Integration		X													
Req.3.2.2 Ground System Integration														X	
Req.4 Environmental Requirements															X
Req.4.1 Space Environment				X											X
Req.4.1.1 Space Environment Survivability				X											
Req.4.2 Earth Environment															
Req.4.2.1 Ground Impact				X											
Req.4.2.2 Sustainability															X

Table 3: Requirements and System Decomposition Matrix

The system level decomposition (Figure 14) and primary subsystem decomposition (Figure 15) will assist in the development of the WBS of the cost estimate and schedule for this program. At this stage, cost estimators will identify cost estimating methodologies which align with this WBS as well as necessary system model data which can be used inputs to these methodologies. This will be an iterative process, as system designers may have yet to define system characteristics at a level necessary to inform the chosen methodologies or cost estimators may need to adjust methodologies to account for costs at a less-detailed view of the system. These characteristics can be defined in the system model, maintaining traceability to system decomposition and requirements, and exported as a report for use in the cost model. Table 4 lists out the necessary system characteristics output from the system model.



System Model & Cost/Schedule Model Interface				
Input Name	**Number**	**Value**	**Units**	**WBS Traceability**
Cost Model Inputs				
PM Headcount	FTE.1	20.00	FTE	*1.1.1.3, 1.2.1.3, 1.3.1*
SEIT Headcount	FTE.2	3.00	FTE	*1.1.1.1, 1.2.1.3, 1.3.1*
Spacecraft Structure/Body - WeightperUnitKg	SysChar.1.1.1.1.1	169.00	KG	*1.1.2.1.1, 1.2.2.1.1*
Spacecraft Structure/Body - StructureMatl	SysChar.1.1.1.1.2	1.40	NA	*1.1.2.1.1*
Spacecraft Structure/Body - LEOEnvironment	SysChar.1.1.1.1.3	1.74	NA	*1.2.2.1.1*
Spacecraft Structure/Body - SubsysHeritage	SysChar.1.1.1.1.3	3.70	NA	*1.2.2.1.1*
Thermal - WeightperUnit	SysChar.1.1.1.2.1	1,488.40	lbs.	*1.1.2.1.2, 1.2.2.1.2*
Communication - Weight per Unit Kg	SysChar.1.1.1.3.1	21.00	KG	*1.1.2.1.3, 1.2.2.1.3*
Communication - TransmitPwr	SysChar.1.1.1.3.2	22.40	Watts	*1.1.2.1.3, 1.2.2.1.3*
Communication - DirectedMission	SysChar.1.1.1.3.3	1.56	NA	*1.1.2.1.3, 1.2.2.1.3*
Guidance, Navigation, and Control - WeightperUnit	SysChar.1.1.1.4.1	460.00	lbs.	*1.1.2.1.4, 1.2.2.1.4*
Guidance, Navigation, and Control - CCDHorGNC	SysChar.1.1.1.4.2	1.60	NA	*1.1.2.1.4, 1.2.2.1.4*
Guidance, Navigation, and Control - Reusable	SysChar.1.1.1.4.3	1.20	NA	*1.1.2.1.4, 1.2.2.1.4*
Guidance, Navigation, and Control - LYMinus1960	SysChar.1.1.1.4.4	22.30	NA	*1.1.2.1.4*
Power - Weight per Unit	SysChar.1.1.1.5.1	900.00	lbs.	*1.1.2.1.5*
IRIS Recovery Payload - TotalWeightKg	SysChar.1.1.2.1	3,000.00	KG	*1.1.2.2*
IRIS Recovery Payload - MaxPwr	SysChar.1.1.2.2	1,000.00	Watts	*1.1.2.2*
Propulsion - Weight per Unit	SysChar.1.1.3.1	6,428.00	lbs.	*1.1.2.4, 1.2.2.4*
Launch Systems Integration (LSI) - TotalFltSysMass	SysChar.3.1	4,000.00	KG	*1.1.2.3*
Launch System Integration (LSI) - FltSysOrgIndustry	SysChar.3.2	2.20	NA	*1.1.2.3*

Table 4: System Characteristics

These system characterstic reports are then formatted to enable rapid input to the cost model so that when an update is made to the system model which impacts thse characteristics, a new report can easily be generated to inform a cost model update. As the system model increases in the level of detail, aligned with system design, the cost model will evolve in tandem to match this level of detail. Cost methodology may also inform areas of system design refinement necessary in the system model. Alternative system design choices will result in variations in system characteristics that alter overall system costs, enabling rapid trade-off analysis.

Like system characteristics, program risks can also be tracked within the system model, and exported in a repeatable fashion to enable rapid updates of project controls artifacts. These risks can be identified by those involved in system design or within the program management team, and it is expected that a single repository/recurring touchpoint be established between each team to summarize these risks. For demonstration purposes, these risks are tracked within the system model and exported for use by the program manamgement team. Figure 16 shows these tracked risks in a matrix, and Table 5 shows the supporting details of these risks in a risk register.

	Negligible	Minor	Moderate	Serious	Critical
High	🟩	🟨	🟥	🟥	🟥 • Risk.1 Funding Delays
Medium High	🟩 • Risk.6 Solar Flare Interference	🟨	🟨	🟥	🟥
Medium	🟩	🟨	🟨	🟨 • Risk.2 Launch Vehicle Failure • Risk.5 Space Debris Collision	🟥
Medium Low	🟩 • Risk.4 Communication Equipment Failure	🟩 • Risk.3 Spacecraft System	🟩 • Risk.7 Navigation System	🟨 • Risk.8 Propulsion System	🟨
Low	🟩	🟩	🟩	🟩	🟨

Figure 16: IRIS Risk Matrix

System Model & Cost/Schedule Model Interface				
Input Name	**Number**	**Probability (%)**	**Cost Impact ($K)**	**Schedule Impact (Months)**
Risks				
Funding Delays	Risk.1	80	$ 100,000	6
Launch Vehicle Failure	Risk.2	40	$ 500,000	3
Spacecraft System Failure	Risk.3	20	$ 200,000	1
Communication Equipment Failure	Risk.4	30	$ 50,000	9
Space Debris Collision	Risk.5	40	$ 1,000,000	2
Solar Flare Interference	Risk.6	60	$ 20,000	0
Navigation System Failure	Risk.7	20	$ 150,000	2
Propulsion System Failure	Risk.8	30	$ 300,000	4

Table 5: IRIS Risk Register

Risk events in the risk register can be formatted in a similar fashion as the system characteristics, in a way that enables rapid export to risk modeling tools. This case study will utilize Joint Confidence Level Analysis (JCLA) to incorporate system characterics/methodologies with an overaching program schedule inclusive of uncertainty and risks.

Developing the Cost Model

In alignment with the pre-planning outlined in Figure 11, the cost estimator will begin constructing a cost estimate based on system inputs contained in the system model. Iterative feedback between both sets of analysts is required so that the cost model will align with the system description in the MBSE tool, and the system model will additionally contain and export information that is relevant for the cost model.

Following from the system decompositions shown in Figure 14 and Figure 15, the cost estimator began developing a program WBS to organize elements of cost in the program which align to the major system elements. Then the cost estimator collected and derived cost estimating methodologies which align to these WBS elements and identified the necessary system characteristics which inform these methodologies. This information was relayed to the systems engineer to specify these system characteristics within the model. System characteristics were then exported in a repeatable report/process for input to the cost model. Certain cost input variables which cannot be derived from the system model were also identified and derived from the remaining available data sources, such as market research or analogies to other projects.

A cost estimating methodology matrix (CEMM) has been prepared to align the IRIS WBS cost elements with the corresponding cost estimating methodologies. Input variables to these equations are aligned with source data: from the system model, market research analysis, or intermediate calculations within the cost model. The CEMM for IRIS is shown in Table 6.

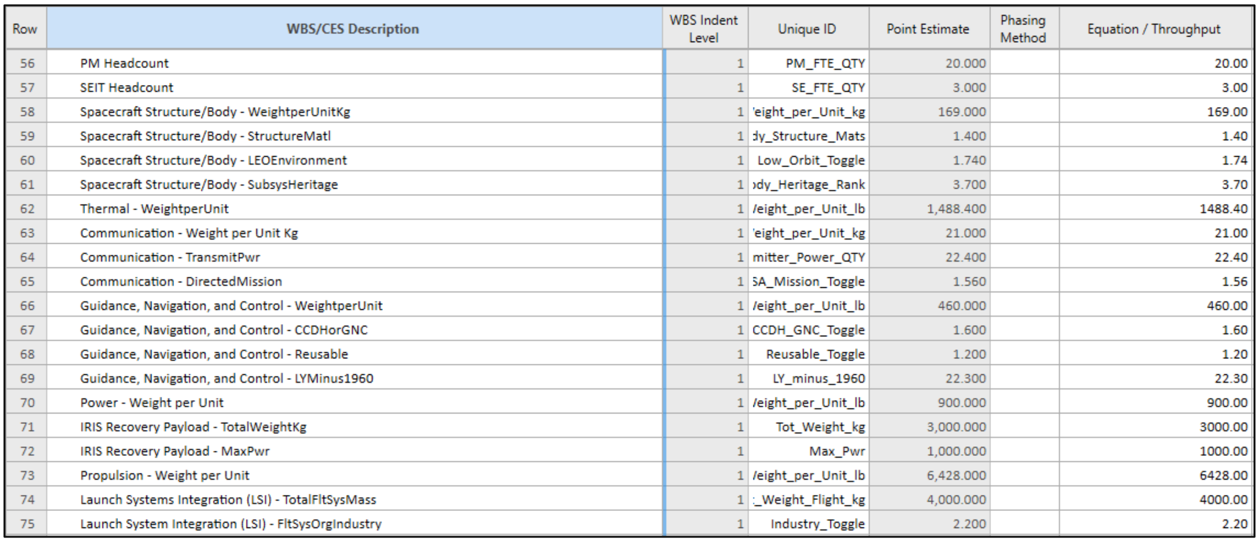
IRIS Program LCCE Cost Estimating Methodology Matrix

WBS #	WBS Element	Equation	Variables Required
1	**IRIS Space System Program**		**NA**
1.1	**IRIS Engineering, Manufacturing & Development (EMD)**		**NA**
1.1.1	SEIT/PM and Support Equipment		NA
1.1.1.1	Systems Engineering	FTEs x Labor Rate	Headcount Labor Rate
1.1.1.2	Assembly, Integration and Test	0.17 * [CostBase]^0.879	CostBase
1.1.1.3	Program Management	FTEs x Labor Rate	Headcount Labor Rate
1.1.2	Space Vehicle		NA
1.1.2.1	Bus		NA
1.1.2.1.1	Spacecraft Structure/Body	35.642 * [WeightPerUnitKg]^0.531 * [SubsysDesignTime]^0.52 * [StructureMatl]^0.721	Weight per Unit Kg SubsystDesignTime StructureMatl
1.1.2.1.2	Thermal Subsystem	0.056 * [WeightPerUnit]^0.784	WeightperUnit
1.1.2.1.3	Communication	178.493 * [WeightPerUnitKg]^0.595 * [TransmitPwr]^0.282 * [DirectedMission]^0.893	Weight per Unit Kg TransmitPwr DirectedMission
1.1.2.1.4	Guidance, Navigation, and Control	60.29 * [WeightPerUnit]^0.343 * [CCDHorGNC]^1.234 * [Reusable]^1.758 * [LYMinus1960]^-1.375	WeightperUnit CCDHonGNC Reusable LYMinus1960
1.1.2.1.5	Power	16.168 + 0.12 * [WeightPerUnit]	WeightPerUnit
1.1.2.2	IRIS Recovery Payload	974 * [TotalWeightKg]^0.54 * [MaxPwr]^0.41	TotalWeightKg MaxPwr
1.1.2.3	Launch Systems Integration (LSI)	31.896 * [TotalFltSysMassKg]^0.495 * [DesignDuration]^0.285 * [FltSysOrgIndustry]^0.722	TotalFltSysMassKg Design Duration FltSysOrgIndustry
1.1.2.4	Propulsion	(0.478 * [WeightPerUnit]^0.641) * Number of Flights	WeightPerUnit
1.1.3	Ground Segment	$15M Trapezoidal Spread to end of development	NA
1.2	**IRIS Production & Deployment**		**NA**
1.2.1	SEIT/PM and Support Equipment		NA
1.2.1.1	Systems Engineering	FTEs x Labor Rate	Headcount Labor Rate
1.2.1.2	Assembly, Integration and Test	0.82 * [CostBase]^0.683	CostBase
1.2.1.3	Program Management	FTEs x Labor Rate	Headcount Labor Rate
1.2.2	Space Vehicle		NA
1.2.2.1	Bus		NA
1.2.2.1.1	Spacecraft Structure/Body	180.503 * [WeightPerUnitKg]^0.617 * [LEOEnvironment]^-0.876 * [SubsysHeritage]^-0.516	Weight per Unit KG LEOEnvironment SubsystHeritage
1.2.2.1.2	Thermal Subsystem	0.0013 * [WeightPerUnit]^1.13	WeightperUnit
1.2.2.1.3	Communication	278.56 * [WeightPerUnitKg]^0.766 * [TransmitPwr]^0.321 * [DirectedMission]^0.862	Weight per Unit Kg TransmitPwr DirectedMission
1.2.2.1.4	Guidance, Navigation, and Control	0.021 * [WeightPerUnit]^0.971 * [CCDHorGNC]^1.015 * [Reusable]^1.513	WeightperUnit CCDHonGNC Reusable
1.2.2.1.5	Power	(0.007 * [WeightPerUnit]) * Number of Flights	WeightPerUnit
1.2.2.2	IRIS Recovery Payload	60% of Total IRIS Payload Design & Development	IRIS Payload DD
1.2.2.4	Propulsion	0.015 * [WeightPerUnit]^0.954	WeightPerUnit
1.2.3	Ground Segment	$7.5M Trapezoidal Spread till First Launch	NA
1.3	**IRIS Sustainment**		**NA**
1.3.1	SEIT/PM and Support Equipment	25% of Avg SEIT/PM Annual Cost	SEIT/PM Cost
1.3.2	Recovery & Retrieval Operation	Cost per Ground Retrieval * Number of Flights	Cost per Ground Retrieval Number of Flights
1.3.3	Maintenance	5% of Total Development Cost, Annually	Total Development Cost
1.3.4	Sustaining Support	10% of Total Production & Deployment Cost, Annually	Total Production Cost

Red = Variable Derived from System Model **Blue** = Variable derived from Market Research or Schedule **Green** = Variable derived from intermediate cost model calculation

Table 6: IRIS Cost Estimating Methodology Matrix

This CEMM serves as the blueprint for the cost model, where system characteristics corresponding to estimate input variables can be easily imported from system model reports. Figure 17 is a screenshot of the cost model which takes in the system variables exported from the system model in Table 4. The remainder of the model aligns to the CEMM shown in Table 6, and references these specific variables within the cost model.

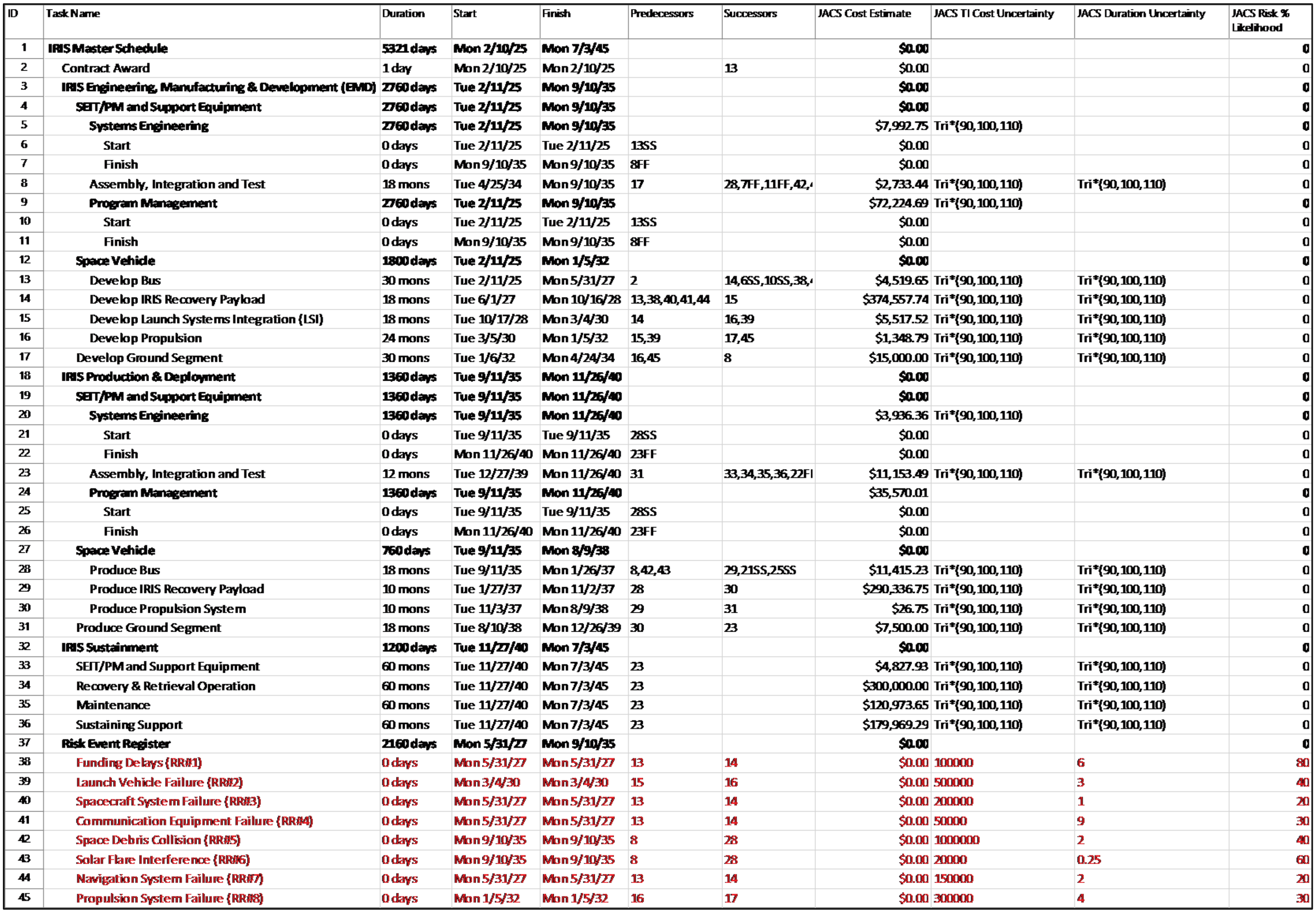


Row	WBS/CES Description	WBS Indent Level	Unique ID	Point Estimate	Phasing Method	Equation / Throughput
56	PM Headcount	1	PM_FTE_QTY	20.000		20.00
57	SEIT Headcount	1	SE_FTE_QTY	3.000		3.00
58	Spacecraft Structure/Body - WeightperUnitKg	1	eight_per_Unit_kg	169.000		169.00
59	Spacecraft Structure/Body - StructureMatl	1	dy_Structure_Mats	1.400		1.40
60	Spacecraft Structure/Body - LEOEnvironment	1	Low_Orbit_Toggle	1.740		1.74
61	Spacecraft Structure/Body - SubsysHeritage	1	dy_Heritage_Rank	3.700		3.70
62	Thermal - WeightperUnit	1	eight_per_Unit_lb	1,488.400		1488.40
63	Communication - Weight per Unit Kg	1	eight_per_Unit_kg	21.000		21.00
64	Communication - TransmitPwr	1	mitter_Power_QTY	22.400		22.40
65	Communication - DirectedMission	1	A_Mission_Toggle	1.560		1.56
66	Guidance, Navigation, and Control - WeightperUnit	1	eight_per_Unit_lb	460.000		460.00
67	Guidance, Navigation, and Control - CCDHorGNC	1	CCDH_GNC_Toggle	1.600		1.60
68	Guidance, Navigation, and Control - Reusable	1	Reusable_Toggle	1.200		1.20
69	Guidance, Navigation, and Control - LYMinus1960	1	LY_minus_1960	22.300		22.30
70	Power - Weight per Unit	1	eight_per_Unit_lb	900.000		900.00
71	IRIS Recovery Payload - TotalWeightKg	1	Tot_Weight_kg	3,000.000		3000.00
72	IRIS Recovery Payload - MaxPwr	1	Max_Pwr	1,000.000		1000.00
73	Propulsion - Weight per Unit	1	eight_per_Unit_lb	6,428.000		6428.00
74	Launch Systems Integration (LSI) - TotalFltSysMass	1	_Weight_Flight_kg	4,000.000		4000.00
75	Launch System Integration (LSI) - FltSysOrgIndustry	1	Industry_Toggle	2.200		2.20

Figure 17: IRIS Cost Model Inputs

By enabling a rapid system characteristic report that can be imported into the cost model, it will be less time-consuming for estimators to update their cost models. For example, if the IRIS Recovery Payload mass decreases in the system model due to the use of lower density materials, the cost model can be updated quickly to show cost impacts. This can reduce the time it takes to provide decision makers with cost relevant information and evaluate trade-offs between system design/performance and the cost.

As the system design matures, increased detail in the system description will emerge which will drive forward increasing detail within the cost model. The WBS will increase in depth, system decompositions will drill down to detailed subassemblies and components, and cost methodologies will need to evolve to reflect this new level of system definition. In this early stage estimate, CERs are primarily used. Future iterations of the estimate will rely more heavily on engineering build-ups, detailed planning of events and design milestones, specific component materials and quantities, and eventually extrapolation of cost actuals.



ID	Task Name	Duration	Start	Finish	Predecessors	Successors	JACS Cost Estimate	JACS TI Cost Uncertainty	JACS Duration Uncertainty	JACS Risk % Likelihood
1	**IRIS Master Schedule**	**5321 days**	**Mon 2/10/25**	**Mon 7/3/45**			**$0.00**			**0**
2	Contract Award	1 day	Mon 2/10/25	Mon 2/10/25		13	$0.00			0
3	**IRIS Engineering, Manufacturing & Development (EMD)**	**2760 days**	**Tue 2/11/25**	**Mon 9/10/35**			**$0.00**			**0**
4	**SEIT/PM and Support Equipment**	**2760 days**	**Tue 2/11/25**	**Mon 9/10/35**			**$0.00**			**0**
5	**Systems Engineering**	**2760 days**	**Tue 2/11/25**	**Mon 9/10/35**			$7,992.75	Tri*(90,100,110)		**0**
6	Start	0 days	Tue 2/11/25	Tue 2/11/25	13SS		$0.00			0
7	Finish	0 days	Mon 9/10/35	Mon 9/10/35	8FF		$0.00			0
8	Assembly, Integration and Test	18 mons	Tue 4/25/34	Mon 9/10/35	17	28,7FF,11FF,42,	$2,733.44	Tri*(90,100,110)	Tri*(90,100,110)	0
9	**Program Management**	**2760 days**	**Tue 2/11/25**	**Mon 9/10/35**			$72,224.69	Tri*(90,100,110)		**0**
10	Start	0 days	Tue 2/11/25	Tue 2/11/25	13SS		$0.00			0
11	Finish	0 days	Mon 9/10/35	Mon 9/10/35	8FF		$0.00			0
12	**Space Vehicle**	**1800 days**	**Tue 2/11/25**	**Mon 1/5/32**			**$0.00**			**0**
13	Develop Bus	30 mons	Tue 2/11/25	Mon 5/31/27	2	14,6SS,10SS,38,	$4,519.65	Tri*(90,100,110)	Tri*(90,100,110)	0
14	Develop IRIS Recovery Payload	18 mons	Tue 6/1/27	Mon 10/16/28	13,38,40,41,44	15	$374,557.74	Tri*(90,100,110)	Tri*(90,100,110)	0
15	Develop Launch Systems Integration (LSI)	18 mons	Tue 10/17/28	Mon 3/4/30	14	16,39	$5,517.52	Tri*(90,100,110)	Tri*(90,100,110)	0
16	Develop Propulsion	24 mons	Tue 3/5/30	Mon 1/5/32	15,39	17,45	$1,348.79	Tri*(90,100,110)	Tri*(90,100,110)	0
17	Develop Ground Segment	30 mons	Tue 1/6/32	Mon 4/24/34	16,45	8	$15,000.00	Tri*(90,100,110)	Tri*(90,100,110)	0
18	**IRIS Production & Deployment**	**1360 days**	**Tue 9/11/35**	**Mon 11/26/40**			**$0.00**			**0**
19	**SEIT/PM and Support Equipment**	**1360 days**	**Tue 9/11/35**	**Mon 11/26/40**			**$0.00**			**0**
20	**Systems Engineering**	**1360 days**	**Tue 9/11/35**	**Mon 11/26/40**			$3,936.36	Tri*(90,100,110)		**0**
21	Start	0 days	Tue 9/11/35	Tue 9/11/35	28SS		$0.00			0
22	Finish	0 days	Mon 11/26/40	Mon 11/26/40	23FF		$0.00			0
23	Assembly, Integration and Test	12 mons	Tue 12/27/39	Mon 11/26/40	31	33,34,35,36,22F	$11,153.49	Tri*(90,100,110)	Tri*(90,100,110)	0
24	**Program Management**	**1360 days**	**Tue 9/11/35**	**Mon 11/26/40**			$35,570.01			**0**
25	Start	0 days	Tue 9/11/35	Tue 9/11/35	28SS		$0.00			0
26	Finish	0 days	Mon 11/26/40	Mon 11/26/40	23FF		$0.00			0
27	**Space Vehicle**	**760 days**	**Tue 9/11/35**	**Mon 8/9/38**			**$0.00**			**0**
28	Produce Bus	18 mons	Tue 9/11/35	Mon 1/26/37	8,42,43	29,21SS,25SS	$11,415.23	Tri*(90,100,110)	Tri*(90,100,110)	0
29	Produce IRIS Recovery Payload	10 mons	Tue 1/27/37	Mon 11/2/37	28	30	$290,336.75	Tri*(90,100,110)	Tri*(90,100,110)	0
30	Produce Propulsion System	10 mons	Tue 11/3/37	Mon 8/9/38	29	31	$26.75	Tri*(90,100,110)	Tri*(90,100,110)	0
31	Produce Ground Segment	18 mons	Tue 8/10/38	Mon 12/26/39	30	23	$7,500.00	Tri*(90,100,110)	Tri*(90,100,110)	0
32	**IRIS Sustainment**	**1200 days**	**Tue 11/27/40**	**Mon 7/3/45**			**$0.00**			**0**
33	SEIT/PM and Support Equipment	60 mons	Tue 11/27/40	Mon 7/3/45	23		$4,827.93	Tri*(90,100,110)	Tri*(90,100,110)	0
34	Recovery & Retrieval Operation	60 mons	Tue 11/27/40	Mon 7/3/45	23		$300,000.00	Tri*(90,100,110)	Tri*(90,100,110)	0
35	Maintenance	60 mons	Tue 11/27/40	Mon 7/3/45	23		$120,973.65	Tri*(90,100,110)	Tri*(90,100,110)	0
36	Sustaining Support	60 mons	Tue 11/27/40	Mon 7/3/45	23		$179,969.29	Tri*(90,100,110)	Tri*(90,100,110)	0
37	**Risk Event Register**	**2160 days**	**Mon 5/31/27**	**Mon 9/10/35**			**$0.00**			**0**
38	Funding Delays (RR#1)	0 days	Mon 5/31/27	Mon 5/31/27	13	14	$0.00	100000	6	80
39	Launch Vehicle Failure (RR#2)	0 days	Mon 3/4/30	Mon 3/4/30	15	16	$0.00	500000	3	40
40	Spacecraft System Failure (RR#3)	0 days	Mon 5/31/27	Mon 5/31/27	13	14	$0.00	200000	1	20
41	Communication Equipment Failure (RR#4)	0 days	Mon 5/31/27	Mon 5/31/27	13	14	$0.00	50000	9	30
42	Space Debris Collision (RR#5)	0 days	Mon 9/10/35	Mon 9/10/35	8	28	$0.00	1000000	2	40
43	Solar Flare Interference (RR#6)	0 days	Mon 9/10/35	Mon 9/10/35	8	28	$0.00	20000	0.25	60
44	Navigation System Failure (RR#7)	0 days	Mon 5/31/27	Mon 5/31/27	13	14	$0.00	150000	2	20
45	Propulsion System Failure (RR#8)	0 days	Mon 1/5/32	Mon 1/5/32	16	17	$0.00	300000	4	30

Figure 18: IRIS JCLA Model

With the cost model and WBS established, the uncertainty-adjusted cost output can be further incorporated into a JCLA model that aligns these costs within an Integrated Master Schedule (IMS) that accounts for uncertainty and risk impacts for the costs and schedule durations. For this case study, the risk events from the risk register (Table 5) are exported from the system model and imported into the JCLA model shown in Figure 18. These risk events are aligned to costs and tasks in the JCLA model so that joint cost and schedule output can be generated. This enables calculations of cost and schedule that accounts for variability in these outputs, as well as account for the impacts of these tracked risks to also inform levels of contingency and management reserve.

These Monte-Carlo cost and schedule outputs are shown in Figure 19 (which shows a scatter plot of finish date against the total cost) and in and Figure 20 (which shows the probability density and cumulative density functions). All of these outputs are created with data imported from the system model and cooperation between the systems engineering team and the project controls team.

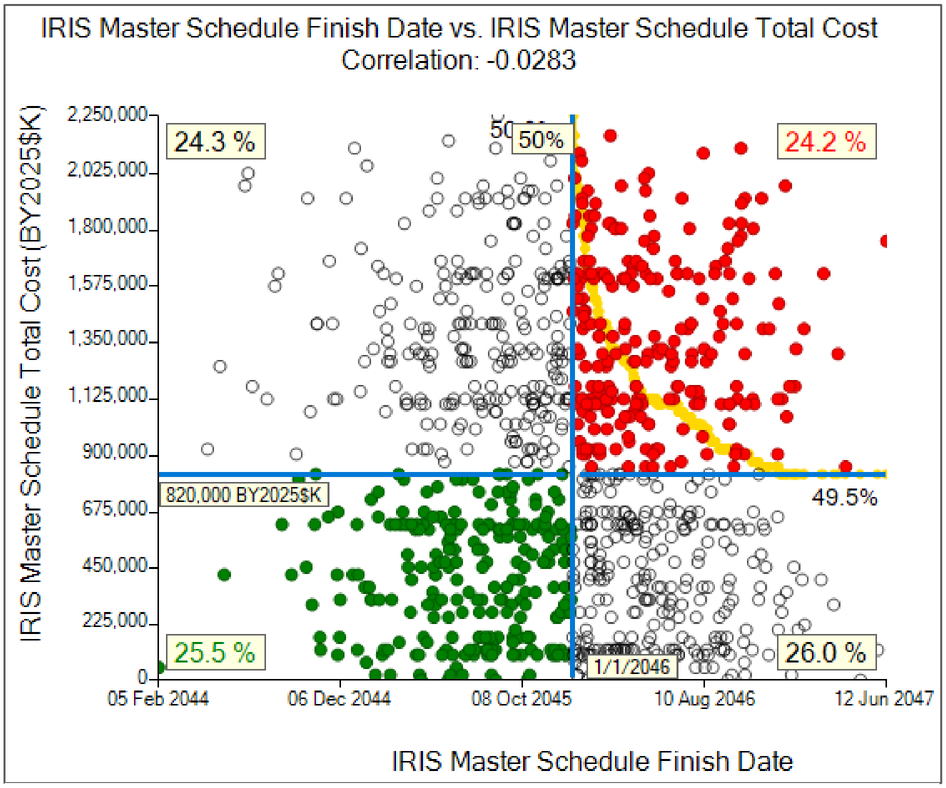


Figure 19: IRIS JCLA Output

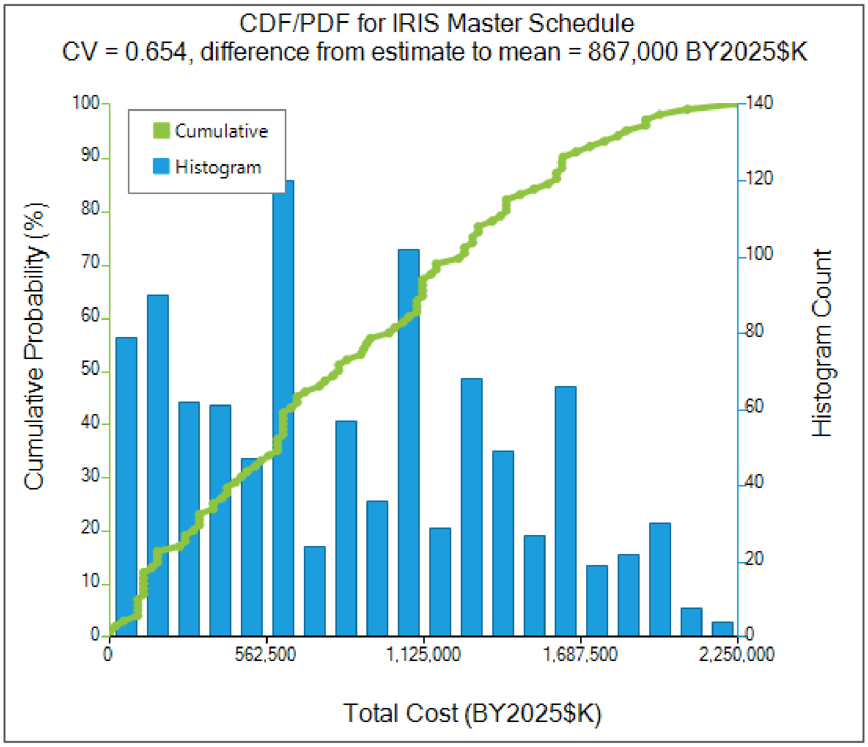


Figure 20: IRIS JCLA Cost Output

Support Through Lifecycle

This case study focused on the system model and cost/schedule output at an early lifecycle stage for this fictional program. Over time, requirements and system design will mature into an increased level of detail. The cost and schedule models should evolve alongside the system model, requiring consistent maintenance of model touch points to ensure interoperability. Touchpoints between each model set should either be updated by both teams manually at defined system design milestones or semi-autonomously in a dynamic fashion whenever substantial maturation of the system design is achieved. The value trade-off for project controls practitioners is that more time will be used maintaining these touch points but there will be increased communication with system designers, easier access to relevant system data, and the ability to produce outputs faster to help inform decision-makers. Additionally, the resulting capability acquired by customers will generally be more well-documented, higher performing, and less likely to be procured at cost-overruns.

Summary

MBSE practitioners and those involved in the field of cost estimation share many similarities in terms of methodology, heuristics, and overarching goals. These two sets of professionals are seldom integrated in a meaningful fashion, despite these similarities. It is mutually beneficial for both disciplines to operate in an operationally integrated environment to increase overall programmatic value to stakeholders as well as further their independent goals beyond what is feasible in a disparate working environment. This requires communication between both disciplines, development and refinement of touchpoints between model sets, and a mutual baseline level of understanding of the opposing discipline. For cost estimators, this requires increasing community-wide understanding of not just MBSE, but general DE practices. Systems engineers, similarly, can familiarize themselves with cost estimating methods and best practices so that developed system models will be more insightful for project controls professionals.

Appendix A: IRIS System Description

Executive Summary: International Recycling In-Orbit System (IRIS)

The International Recycling In-Orbit System (IRIS) is an innovative program designed to address the growing problem of space debris. As the number of satellites and space missions increases, so does the accumulation of debris in Earth's orbit, posing significant risks to active satellites, space stations, and future missions. IRIS aims to mitigate these risks by implementing a sustainable and efficient solution for debris collection and recycling.

Program Overview: IRIS utilizes advanced heat shield capsules to capture and safely transport space debris from orbit back to Earth. These capsules are equipped with state-of-the-art technology to identify, collect, and contain debris of various sizes and materials. Once filled, the capsules re-enter Earth's atmosphere, protected by their heat shields, and land in designated recovery zones.

Key Objectives:

1. **Reduce Space Debris:** By actively removing debris from orbit, IRIS helps to maintain a safer space environment for current and future missions.
2. **Promote Sustainability:** The collected debris is transported to Earth, where it undergoes recycling processes to recover valuable materials, reducing the need for new raw materials and minimizing environmental impact.
3. **Enhance Safety:** By decreasing the amount of debris in orbit, IRIS reduces the risk of collisions that can damage operational satellites and spacecraft.

4. **Economic Efficiency:** Recycling in-orbit debris provides a cost-effective alternative to manufacturing new materials, supporting the space industry's economic sustainability.

Technological Innovation: IRIS capsules are designed with cutting-edge heat shield technology that ensures safe re-entry and landing. The system's advanced sensors and robotic arms enable precise debris collection, while its modular design allows for scalability and adaptability to different mission requirements.

Strategic Partnerships: IRIS collaborates with international space agencies, private aerospace companies, and environmental organizations to leverage expertise, share resources, and promote global cooperation in space sustainability efforts.

Conclusion: The International Recycling In-Orbit System (IRIS) represents a groundbreaking approach to managing space debris. By combining innovative technology with a commitment to sustainability, IRIS not only addresses a critical challenge in space exploration but also sets a new standard for environmental stewardship in the aerospace industry.

Requirements for the International Recycling In-Orbit System (IRIS)

Introduction

The International Recycling In-Orbit System (IRIS) aims to address the issue of space debris by collecting and transporting it back to Earth for recycling. This document outlines the requirements for the development, deployment, and operation of the IRIS program.

System Overview

IRIS will utilize heat shield capsules equipped with advanced technology to capture, contain, and safely return space debris to Earth. The system will include ground-based facilities for processing and recycling the collected materials.

Functional Requirements

Req 1.1 Debris Detection and Collection

- **1.1.1** The system shall be capable of detecting space debris of various sizes, from small fragments to larger objects.
- **1.1.2** The system shall include robotic arms or similar mechanisms to capture and secure debris.
- **1.1.3** The system shall be able to operate autonomously or be remotely controlled by ground operators.

Req 1.2 Capsule Design and Operation

- **1.2.1** Capsules shall be equipped with heat shields to protect against re-entry temperatures.
- **1.2.2** Capsules shall have a modular design to allow for scalability and adaptability to different mission requirements.

- **1.2.3** Capsules shall include sensors and telemetry systems to monitor and report status during the mission.

Req 1.3 Re-entry and Recovery

- **1.3.1** Capsules shall be designed to ensure safe re-entry and landing in designated recovery zones.
- **1.3.2** The system shall include mechanisms for precise landing to facilitate easy recovery.
- **1.3.3** Capsules shall be equipped with tracking devices to enable location and retrieval after landing.

Req 1.4 Ground Processing and Recycling

- **1.4.1** Ground facilities shall be equipped to handle and process various types of materials collected from space debris.
- **1.4.2** The recycling process shall comply with environmental regulations and standards.
- **1.4.3** The system shall include protocols for the safe handling and disposal of hazardous materials.

Performance Requirements

Req 2.1 Efficiency

- **2.1.1** The system shall be capable of collecting a minimum of 90% of targeted debris within a specified mission duration.
- **2.1.2** The system shall minimize fuel consumption and operational costs.

Req 2.2 Reliability

- **2.2.1** The system shall have a reliability rate of at least 95% for successful debris collection and return missions.
- **2.2.2** The system shall include redundancy features to ensure mission success in case of component failure.

Req 2.3 Safety

- **2.3.1** The system shall comply with all relevant safety standards for space operations.
- **2.3.2** The system shall include fail-safe mechanisms to prevent accidental release of debris during collection and transport.

Interface Requirements

Req 3.1 Communication

- **3.1.1** The system shall include secure communication channels for data transmission between the capsules and ground control.
- **3.1.2** The system shall support real-time monitoring and control capabilities.

Req 3.2 Integration

- **3.2.1** The system shall be compatible with existing space infrastructure and launch vehicles.
- **3.2.2** The system shall include interfaces for integration with ground processing and recycling facilities.

Environmental Requirements

Req 4.1 Space Environment

- **4.1.1** The system shall be designed to operate in the harsh conditions of space, including extreme temperatures, radiation, and microgravity.

Req 4.2 Earth Environment

- **4.2.1** The system shall ensure minimal environmental impact during re-entry and landing.
- **4.2.2** The recycling process shall aim to reduce the carbon footprint and promote sustainability.

Conclusion

The IRIS program represents a significant step towards sustainable space operations by addressing the critical issue of space debris. By meeting the outlined requirements, IRIS will enhance the safety and sustainability of space activities while promoting environmental stewardship.

Appendix B: Bibliography

[1] Office of the Under Secretary of Defense, Research and Engineering (OUSD(R&E)), "Digital Engineering, Modeling & Simulation," 2025. [Online]. Available: https://www.cto.mil/sea/dems/.

[2] L. E. Hart, "Introduction To Model-Based System Engineering (MBSE) and SysML," 30 July 2015. [Online]. Available: https://www.incose.org/docs/default-source/delaware-valley/mbse-overview-incose-30-july-2015.pdf.

[3] W. Gruhl, "Relation of SE Investments to NASA Program Cost Overruns," in *NASA*, 1992.

[4] A. Bernal, "Engineering Cheatsheet," 7 March 2022. [Online]. Available: https://engineeringcheatsheet.com/describe-10-characteristics-of-the-automotive-product-development-process-illustrated-in-the-systems-engineering-v-model/.

[5] Government Accountability Office, "GAO," March 2020. [Online]. Available: https://www.gao.gov/assets/d20195G.pdf.

[6] ICEAA, "ICEAA Archive," [Online]. Available: https://www.iceaaonline.com/archives/.

[7] INCOSE, "INCOSE Library," [Online]. Available: https://www.incose.org/incose-content-library/incose-content-library.

[8] Cambridge English Dictionary, "Cambridge Dictionary," Cambridge, [Online]. Available: https://dictionary.cambridge.org/us/dictionary/english/process.

[9] Defense Acquisition University (DAU), "DAU," [Online]. Available: https://www.dau.edu/acquipedia-article/cost-analysis-requirements-description-card.

Appendix C: Acronym Table

Acronym/Abbreviation	Definition
ASoT	Authoritative Source of Truth
CARD	Cost Analysis Requirements Document
CDF	Cumulative Density Function
CEMM	Cost Estimating Methodology Matrix
CER	Cost Estimating Relationships
DE	Digital Engineering
DoD	Department of Defense
GAO	Government Accountability Office
ICEAA	International Cost Estimating & Analysis Association
IMS	Integrated Master Schedule
INCOSE	International Council on Systems Engineering
IRIS	International Recycling In-Orbit System
JCLA	Joint Confidence Level Analysis
MBSE	Model-Based System Engineering
PDF	Probability Density Function
WBS	Work Breakdown Structure