

**Appendix A to “Understanding Organizational Factors in the Aerospace Industry’s Transition to Model-Based Systems Engineering,” submitted for publication to the *INCOSE Journal of Systems Engineering*, May 2025**

Sarah G. Shaw  
PhD Candidate  
Department of Systems Engineering, Colorado State University

*Appendix A*

**A.1 System Requirements Definition and Traceability**

The first Systems Engineering (SE) project defined for this research is System Requirements Definition and Traceability. The tasks for this project were initially drafted using the International Council on Systems Engineering (INCOSE) Systems Engineering Handbook v5<sup>1</sup> with review and updates from the Digital Engineering (DE) and SE expert panel. The following table shows the finalized list of tasks for this project.

*Table A.1. Tasks for System Requirements Definition and Traceability*

<i>Project 1: System Requirements Definition and Traceability</i>	
<i>Task 1: Program Management/Admin</i>	
1.1.1	Identify stakeholders and manage engagement, expectations, and needs
1.1.2	Define program/system scope, including system boundaries and work breakdown structure
1.1.3	Define and manage program/acquisition strategy and implement business rules
1.1.4	Define and drive vision and way-ahead for the team
1.1.5	Manage project progress and outcome, including understanding the programmatic/technical/cost/schedule implications of programmatic/technical/cost/schedule decisions
1.1.6	Define and manage cost/budget
1.1.7	Define and manage schedule, ensuring focus on major milestones
1.1.8	Define and manage personnel, roles, teams, and organizational structure, to include internal teaming agreements/needs, meeting cadence, reporting/deliverables requirements, and training
1.1.9	Define and manage required tools and equipment, including Information Technology (IT) infrastructure, prototypes, simulators, testbeds, engineering models, etc.
1.1.10	Define and manage risk and program compliance
1.1.11	Manage access to facilities/spaces, including IT infrastructure
1.1.12	Define process for and manage Change Management, Configuration Control (requirements, project documentation, etc), and Quality Control
1.1.13	Define, manage, and execute required reviews
1.1.14	Manage project documentation/technical document library and compile and deliver relevant contract deliverables and documents
1.1.15	Manage Security and maintain Security Classification Guide
1.1.16	Manage IT/cyber support and ensure coordination with Security
<i>Task 2: Define System Requirements</i>	
1.2.1	Define functional boundary in terms of the behavior and properties the system will provide, including system performance attributes, throughput, and responsiveness needs
1.2.2	Identify/obtain stakeholder key functional/performance requirements and iteratively review
1.2.3	Define functions and associated performance, system Concept of Operations (CONOPs), and Use Cases
1.2.4	Define constraints on the program/system, operating environment, interactions with external systems, externally-driven requirements, and required new interfaces; create required documentation
1.2.5	Identify and capture system requirements and attributes and create requirements baseline
1.2.6	Identify requirements that relate to risk, criticality, critical quality characteristics, and compliance with standards and regulations
1.2.7	Begin to define verification success criteria for each requirement, verification strategy, verification method, verification plan, and responsible organization, and continue to iterate
<i>Task 3: Analyze System Requirements</i>	
1.3.1	Analyze requirements to ensure they are correct, complete, consistent, traceable, understandable, appropriate to level, verifiable/measurable, and feasible, adjudicating and refining as necessary
1.3.2	Define critical performance measures
1.3.3	Review analyzed requirements with stakeholders and resolve any issues

1.3.4	Analyze requirements for criticality (e.g., impact of non-satisfaction) and sensitivity (e.g., impact of partial satisfaction)
1.3.5	Prioritize requirements, if needed
1.3.6	Develop performance models and analyses to model and/or test to validate and/or verify requirements
1.3.7	Determine system compliance criteria and margin to requirements
<i>Task 4: Manage System Requirements</i>	
1.4.1	Establish and sustain requirements traceability
1.4.2	Establish, document, and manage system requirements change process, including a means to identify impacts of a proposed requirements change
1.4.3	Establish and manage baselines for information items, work products, or other artifacts
1.4.4	Manage progress of requirements development, change, and satisfaction

## A.2 Pre-Phase A Architecture Development

The task list for Pre-Phase A Architecture Development was drafted from a study by a National Aeronautics and Space Administration (NASA) program for architecting a robotic space system<sup>2</sup> and reviewed by the panel of DE and SE experts. The following table shows the finalized list of tasks for this project.

*Table A.2. Tasks for Pre-Phase A Architecture Development*

<i>Project 2: Pre-Phase A Architecture Development</i>	
<i>Task 1: Program Management/Admin</i>	
1.1.1	Identify stakeholders and manage engagement, expectations, and needs
1.1.2	Define program/system scope, including system boundaries and work breakdown structure
1.1.3	Define and manage program/acquisition strategy and implement business rules
1.1.4	Define and drive vision and way-ahead for the team
1.1.5	Manage project progress and outcome, including understanding the programmatic/technical/cost/schedule implications of programmatic/technical/cost/schedule decisions
1.1.6	Define and manage cost/budget
1.1.7	Define and manage schedule, ensuring focus on major milestones
1.1.8	Define and manage personnel, roles, teams, and organizational structure, to include internal teaming agreements/needs, meeting cadence, reporting/deliverables requirements, and training
1.1.9	Define and manage required tools and equipment, including IT infrastructure, prototypes, simulators, testbeds, engineering models, etc.
1.1.10	Define and manage risk and program compliance
1.1.11	Manage access to facilities/spaces, including IT infrastructure
1.1.12	Define process for and manage Change Management, Configuration Control (requirements, project documentation, etc), and Quality Control
1.1.13	Define, manage, and execute required reviews
1.1.14	Manage project documentation/technical document library and compile and deliver relevant contract deliverables and documents
1.1.15	Manage Security and maintain Security Classification Guide
1.1.16	Manage IT/cyber support and ensure coordination with Security
<i>Task 2: Define System-Level Architecture</i>	
2.2.1	Perform system-level trade study to research and generate possible system architectures and make a recommendation
2.2.2	Conduct system architecture peer review to ensure the appropriateness and potential success of recommended system architecture against key functional/performance requirements; iteratively review with appropriate stakeholder(s)
2.2.3	Capture final system CONOPs and define architecture ownership roles
2.2.4	Identify and elaborate on system Use Cases
2.2.5	Define system structural elements, including boundaries, relationships, interfaces, exchanges, and interactions with elements external to the system
2.2.6	Define system internal structure, including the functions of and relationships, interfaces, exchanges, and interactions between the proposed subsystems of the module
2.2.7	Define system element specifications to include relevant information, such as physical characteristics/constraints, environment, test support features, software features, links to drawings and specifications, etc.

2.2.8	Define system-level data, including the scope, structure, and location of the data, its flow to and from the module, and algorithm needs
2.2.9	Define system-level behavior within the context of its use
2.2.10	Define, map, and allocate system-level requirements, including constraints and interactions; requirements that relate to risk, critical quality characteristics, and compliance; and success criteria
2.2.11	Determine modeling strategy
<i>Task 3: Define Subsystem-Level Architecture (performed for each subsystem)</i>	
2.3.1	Perform subsystem-level trade study to research and generate possible subsystem concepts/architectures and make a recommendation based on ability to meet key requirements within cost/schedule
2.3.2	Conduct subsystem architecture peer review to ensure the appropriateness and potential success of recommended subsystem concepts
2.3.3	Define subsystem structural elements, including boundaries, relationships, interfaces, exchanges, and interactions with elements external to each subsystem
2.3.4	Define subsystem internal structure, including the functions of and relationships, interfaces, exchanges, and interactions between the proposed assemblies of the subsystem
2.3.5	Define subsystem element specifications to include relevant information, such as physical characteristics/constraints, environment, links to drawings and specifications, etc.
2.3.6	Define subsystem-level data, including the scope, structure, and location of the data, its flow to and from each subsystem, and algorithm needs
2.3.7	Define subsystem behavior within the context of its use
2.3.8	Define, map, and allocate subsystem-level requirements, including constraints and interactions; requirements that relate to risk, critical quality characteristics, and compliance; and success criteria
<i>Task 4: Define Assembly-Level Architecture (performed for each assembly that makes up the subsystems, e.g., individual actuators, mechanisms, structural elements, and sensors)</i>	
2.4.1	Perform assembly-level trade study to research and generate possible assembly concepts and make a recommendation
2.4.2	Conduct assembly architecture peer review to ensure the appropriateness and potential success of recommended assembly concepts
2.4.3	Define assembly structural elements, including boundaries, relationships, interfaces, exchanges, and interactions with elements external to each assembly
2.4.4	Define assembly internal structure, including the functions of and relationships, interfaces, exchanges, and interactions between the proposed elements of the assembly
2.4.5	Define assembly element specifications to include relevant information, such as physical characteristics/constraints, environment, links to drawings and specifications, etc.
2.4.6	Define assembly-level data, including the scope, structure, and location of the data, its flow to and from the assembly, and algorithm needs
2.4.7	Define assembly behavior within the context of its use
2.4.8	Define, map, and allocate assembly-level requirements, including constraints and interactions; requirements that relate to risk, critical quality characteristics, and compliance; and success criteria

### A.3 Concept Definition

A third SE project for this research is Concept Definition. This set of tasks was derived from a Work Breakdown Structure (WBS) for a program by an industry development contractor and reviewed and finalized by the panel of SE and DE experts. The following table shows the list of finalized tasks for this project.

*Table A.3. Tasks for Concept Definition*

<i>Project 3: Concept Definition</i>	
<i>Task 1: Program Management/Admin</i>	
1.1.1	Identify stakeholders and manage engagement, expectations, and needs
1.1.2	Define program/system scope, including system boundaries and work breakdown structure
1.1.3	Define and manage program/acquisition strategy and implement business rules
1.1.4	Define and drive vision and way-ahead for the team
1.1.5	Manage project progress and outcome, including understanding the programmatic/technical/cost/schedule implications of programmatic/technical/cost/schedule decisions
1.1.6	Define and manage cost/budget
1.1.7	Define and manage schedule, ensuring focus on major milestones

1.1.8	Define and manage personnel, roles, teams, and organizational structure, to include internal teaming agreements/needs, meeting cadence, reporting/deliverables requirements, and training
1.1.9	Define and manage required tools and equipment, including IT infrastructure, prototypes, simulators, testbeds, engineering models, etc.
1.1.10	Define and manage risk and program compliance
1.1.11	Manage access to facilities/spaces, including IT infrastructure
1.1.12	Define process for and manage Change Management, Configuration Control (requirements, project documentation, etc), and Quality Control
1.1.13	Define, manage, and execute required reviews
1.1.14	Manage project documentation/technical document library and compile and deliver relevant contract deliverables and documents
1.1.15	Manage Security and maintain Security Classification Guide
1.1.16	Manage IT/cyber support and ensure coordination with Security
<i>Task 2: System Analysis and Architecture Design</i>	
3.2.1	Identify and analyze stakeholder system requirements, including constraints and interactions; requirements that relate to risk, critical quality characteristics, and compliance; key performance requirements; and success criteria; decomposing the system requirements into subsystem requirements, as needed. Iteratively review with appropriate stakeholders.
3.2.2	Analyze desired stakeholder system Use Cases, including relationships, interfaces, exchanges, and interactions between the proposed system and its external environment
3.2.3	Perform system-level trade study to research and generate possible system concepts and make a recommendation
3.2.4	Define system functional architecture, relating the functions to system requirements
3.2.5	Allocate system functional architecture to structural elements
3.2.6	Parameterize system functional architecture to include relevant information, such as physical characteristics/constraints, environment, etc.
3.2.7	Develop modeling strategy and model/simulate system behavior
<i>Task 3: Subsystem Analysis and Architecture Design</i>	
3.3.1	Analyze stakeholder subsystem requirements, including constraints and interactions; requirements that relate to risk, critical quality characteristics, and compliance; and success criteria; decomposing the subsystem requirements into component requirements, as needed. Iteratively review with appropriate stakeholders.
3.3.2	Perform subsystem-level trade study to research and generate possible subsystem concepts and make a recommendation
3.3.3	Define subsystem functional architecture, relating the functions to subsystem requirements
3.3.4	Allocate subsystem functional architecture to structural elements
3.3.5	Parameterize subsystem functional architecture to include relevant information, such as physical characteristics/constraints, environment, etc.
3.3.6	Revisit/asses modeling strategy and perform end-to-end modeling, simulation, and analysis through subsystems to ensure both CONOPs closure and likelihood of meeting key performance requirements
<i>Task 4: Component Analysis and First Unit Architecture Design</i>	
3.4.1	Analyze stakeholder component requirements, including constraints and interactions; requirements that relate to risk, critical quality characteristics, and compliance; and success criteria. Iteratively review with appropriate stakeholders.
3.4.2	Perform component-level trade study to research and generate possible component concepts and make a recommendation
3.4.3	Define component functional architecture, relating the functions to component requirements
3.4.4	Allocate component functional architecture to structural elements
3.4.5	Parameterize component functional architecture to include relevant information, such as physical characteristics/constraints, environment, etc.

## REFERENCES

1. Walden DD, Shortell TM, Roedler GJ, et al., eds. *INCOSE Systems Engineering Handbook, 5th Edition*. 5th ed. John Wiley & Sons, Inc.; 2023. Accessed July 17, 2023. <https://www.wiley.com/en-us/INCOSE+Systems+Engineering+Handbook%2C+5th+Edition-p-9781119814290>
2. Younse PJ, Cameron JE, Bradley TH. Comparative Analysis of Model-Based and Traditional Systems Engineering Approaches for Architecting a Robotic Space System Through Automatic Information Transfer. *IEEE Access*. 2021;9:107476-107492. doi:[10.1109/ACCESS.2021.3096468](https://doi.org/10.1109/ACCESS.2021.3096468)