A Brief Discussion on Systems Engineering Process Outputs

Dr. Akhila Udupa
Associate Professor, Masters In Business Administration (General Management), Presidency University, Bangalore, India,
Email Id-akhila.udupa@presidencyuniversity.in

ABSTRACT:

The Systems Engineering Process (SEP) is a methodical process used to design and oversee the life cycle of complex systems. The important outcomes produced throughout the different SEP phases are highlighted in this abstract. The effective design, development, implementation, and operation of systems depend heavily on these outputs. The collecting and analysis of requirements during the first stage of the SEP results in the compilation of a System Requirements Document (SRD). The SRD provides a basis for further development efforts by outlining the functional and performance requirements for the system. An Architectural Design Document (ADD) is created once the system architecture has been developed and the requirements have been clarified. The ADD offers a high-level breakdown of the system's architecture, interfaces, and component interactions. Detailed design efforts start after the system requirements and architecture are established. Outputs from this phase include test plans, interface control papers, and comprehensive design documentation. These papers provide the particulars of the subsystems, interfaces, and testing techniques required for the system's development and integration. The SEP generates verification and validation artefacts as development advances. Verification operations result in the creation of verification plans, test reports, and verification matrices by confirming that the system interfaces and components fulfil the required specifications.

KEYWORDS:

Acquisition Programme Baselines, Interface Specifications, Risk Management, System Architecture.

I. INTRODUCTION

The design, development, and management of complex systems may be done in a systematic and multidisciplinary manner using the systems engineering method. The system is defined, created, and subjected to a broad variety of activities and tasks during this process, producing a number of outputs that direct the creation and execution of the system. The products of the systems engineering process serves as important artefacts and documentation that record the choices, considerations, and system requirements. These results provide users a thorough grasp of the system's requirements, design, interfaces, and functioning, promoting efficient stakeholder collaboration and communication [1], [2].

The following are some of the main outcomes of the systems engineering process:

System Requirements: The discovery and documenting of the system requirements mark the start of the systems engineering process. These specifications contain the needs for the system's functionality, performance standards, limitations, and other elements. The system requirements serve as the cornerstone for all future design and development work [3], [4].

System Architecture: A system architecture that outlines the organisation and structure of the system is created via the systems engineering process. A high-level picture of how the system's components interact to provide the intended functions is provided by the identification of the components, subsystems, and linkages between them [5], [6].

System Design: The systems engineering process produces in-depth system designs based on the system requirements and architecture. These designs include an extensive specification of the system's parts, connections, and capabilities. They describe the system's mechanical, electrical, and physical components and direct the integration and implementation procedures.

Interface Specifications: The systems engineering process produces thorough interface specifications because interfaces are essential to system integration. The communication protocols, mechanical connections, data formats, and other criteria necessary for flawless interaction between system components or subsystems are outlined in these standards [7], [8].

Verification and Validation Plans: The systems engineering process produces verification and validation plans, which describe the approaches and techniques for assessing the system's overall performance and conformance with the requirements. These plans outline the evaluations, testing, and reviews that must be carried out to guarantee that the system performs as planned.

System documentation: A crucial result of the systems engineering process is documentation. It contains a variety of papers, including user manuals, design specifications, system requirements documents, technical reports, and configuration management strategies. These records provide a thorough account of the system's creation, use, and upkeep, simplifying future reference and improvements [9], [10].

Risk Management Plans: As part of the systems engineering process, possible hazards related to the system are identified, and risk management plans are created. These plans include ways to reduce, keep an eye on, and manage the identified risks during the course of the system's lifespan.

These results ensure that system developers, stakeholders, and users have a common understanding of the needs, design, and functioning of the system. They support the effective design and deployment of the system by directing later processes including system implementation, production, testing, and maintenance. Therefore, the judgements, analyses, and system requirements are captured in the outputs of the systems engineering process. They include of risk management strategies, system documentation, architecture, design, interface specifications, verification and validation plans, and system requirements. Through good communication, collaboration, and decision-making among stakeholders throughout the system's lifespan, these outputs provide a thorough knowledge of the system.

II. DISCUSSION

Documenting Requirements and Designs

The papers that outline the system requirements and design solution make up the outputs of the systems engineering process. The system architecture is completed by adding enabling goods and services to the physical architecture that was created during the synthesis phase. The reference model for future system requirements and documentation development is this system level design. The decision database, as well as the system and configuration item designs, specifications, and baselines, are all products of the system engineering process. The degree of development affects outputs. As system definition moves from idea to detailed design, they become more technical. The data created when each step of system definition is completed serves as the input for further applications of the system engineering process.

Architectures: System/Configuration Item

The system as a whole is described in the system architecture. In addition to the enabling goods and services needed for life cycle employment, support, and management, it also includes the physical architecture created via design synthesis. Work Breakdown Structures, a chapter in the Military Handbook (MIL-HDBK)-881, offers reference models for weapon system designs. MIL-HDBK-881 exemplifies the first three layers of conventional system topologies, as shown in Figure 1. MIL-HDBK-881 templates may be used by Programme Offices to create a top-level architecture during the system definition phase that is suited to the requirements of the particular system under consideration. The tiers below these first three will typically be developed by the design contractor.

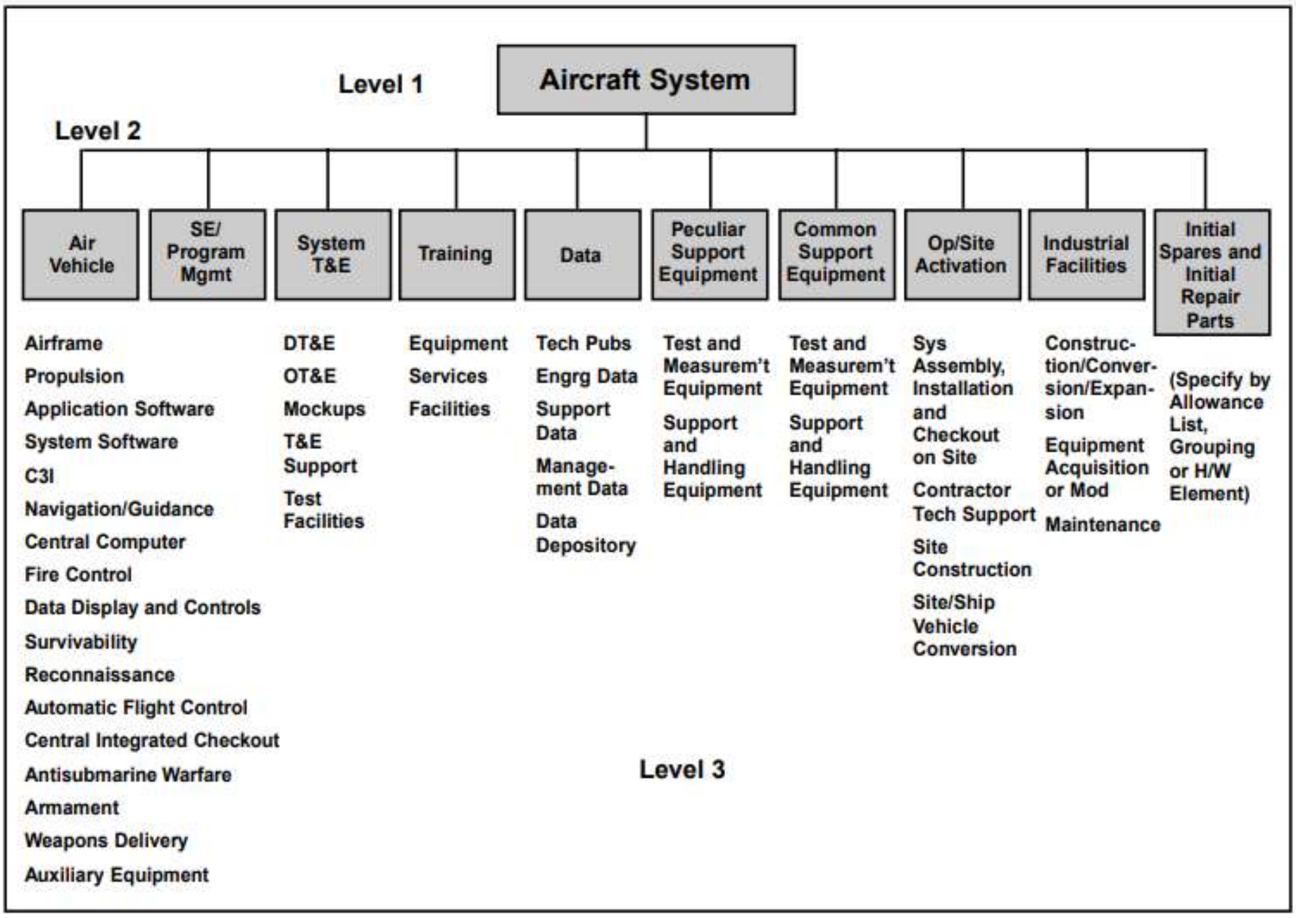

Figure 1: Example from MIL-HDBK-881 [ocw.mit.edu].

Specifications

A specification is a written document that precisely and concisely outlines the fundamental technical requirements for goods, components, or services, as well as the methods by which the requirements may be validated. Specifications reduce duplication and inconsistencies, enable accurate estimates of required work and resources, serve as a negotiation and reference document for engineering changes, provide configuration documentation, and facilitate consistent communication among those in charge of the eight core functions of systems engineering. In order for IPTs to effectively design the system and calculate the cost of design options, they provide them a clear understanding of the issue that has to be addressed. They provide testers instructions on how to verify (qualify) each technical requirement.

Program-Specific Requirements

A number of specifications are created throughout the system development process to define the system at various degrees of detail. The foundation of the configuration baselines is made up of these special programme standards. As shown in Figure 2, these baselines are established at various stages of the design process in addition to referring to various levels within the system hierarchy.

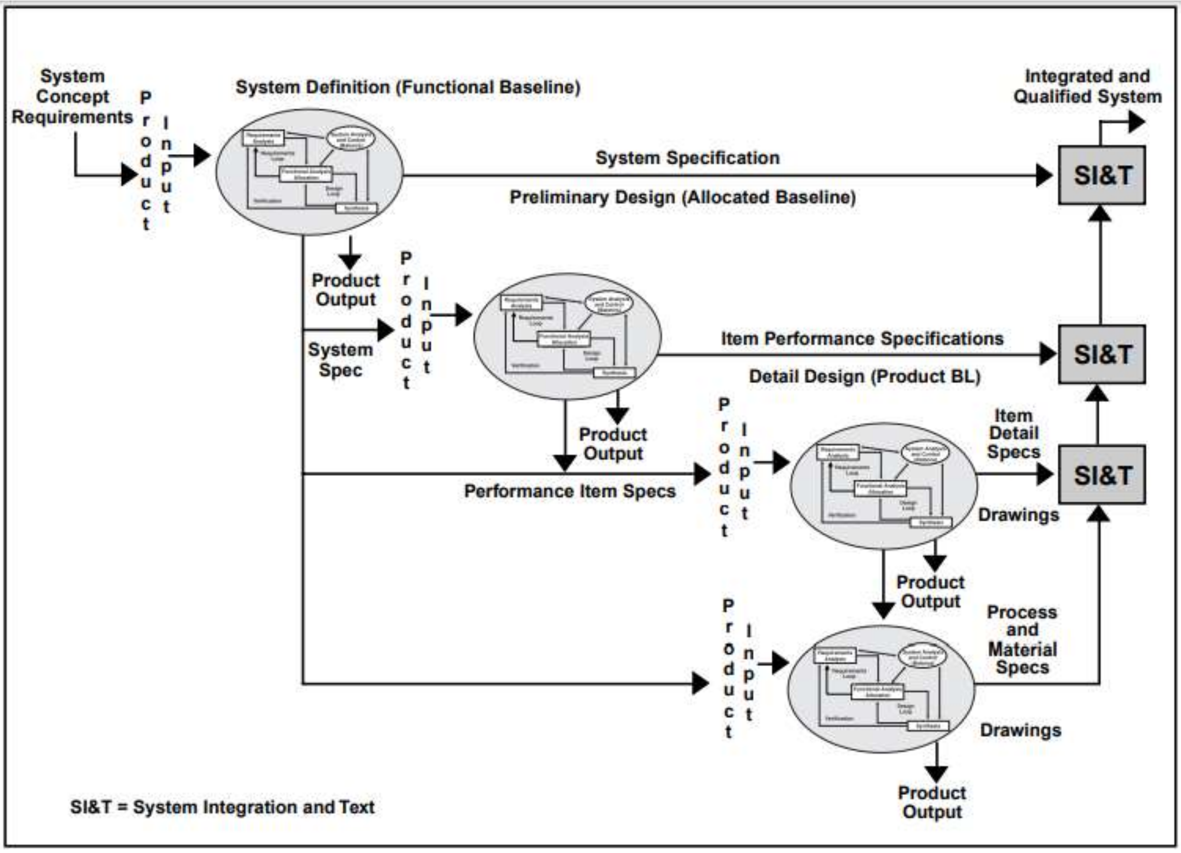

Figure 2: Specifications and Levels of Development.

The system's top-level (system) functions, performance, and interfaces are first outlined. The operational needs specified by the user are the source of these technological requirements. The System Specification, which is the main source of information for the system-level Functional Baseline, contains a description of this system-level technical information. A set of design criteria is then produced for each of the objects that are below the system level as a result of the system requirements being flowed down (assigned) to those elements. The Allocated Baseline (also known as the "Design to" baseline) is documented by a set of Item Performance Specifications, which also include further interface definitions, process descriptions, and drawings. After the basic design specifications for each item have been established, detailed design comes next. Detailing the system in terms of the physical entities that will be used to fulfil the design criteria entails defining the system from top to bottom.

Specification	Content	Baseline
System Spec	Defines mission/technical performance requirements. Allocates requirements to functional areas and defines interfaces.	Functional
Item Performance Spec	Defines performance characteristics of CIs and CSCIs. Details design requirements and with drawings and other documents form the Allocated Baseline.	Allocated "Design To"
Item Detail Spec	Defines form, fit, function, performance, and test requirements for acceptance. (Item, process, and material specs start the Product Baseline effort, but the final audited baseline includes all the items in the TDP.)	Product "Build To" or "As Built"
Process Spec	Defines process performed during fabrication.	
Material Spec	Defines production of raw materials or semi-fabricated material used in fabrication.	

Figure 3: Specifications Types [ocw.mit.edu].

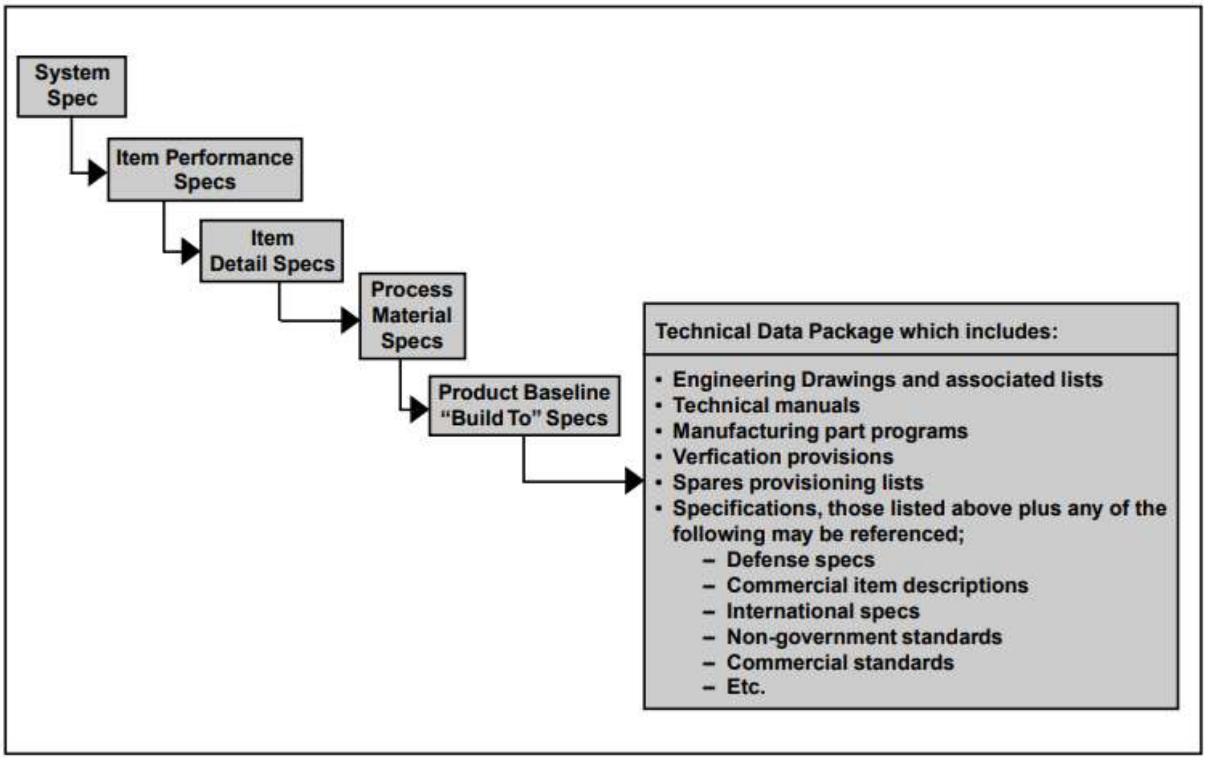

Figure 4: Specifications and Design Documents: A Relationship [ocw.mit.edu].

A final baseline is established when thorough design is finished. According to the development stage, this is often referred to as the Product Baseline and may include a "Build to" or "As built" description. The Technical Data Package, which includes Item Detail Specifications, Process and Material Specifications, as well as drawings, component lists, and other data that fully characterises the finished system in physical detail, serves as documentation for the Product Baseline. Figure 3 depicts the relationship between these requirements and the corresponding baselines.

Role of Specifications

Documents describing requirements explain the need of the development. Specification papers are a middle-ground articulation of the technical requirements (function, performance, and interface) for the required system. Drawings, related lists, and other design papers specify how the design criteria are to be met. The requirement flow from high-level specifications to design documents is shown in Figure 4. The process of system engineering includes creating specifications, but it also involves procedures related to legal and editorial communication abilities. Some general guidelines are shown in Figure 5 to support this. Specifications describe what the system must do, how well it must do it, and how to check that it can.

- **Use a table of contents and define all abbreviations and acronyms.**
- **Use active voice.**
- **Use "shall" to denote mandatory requirement and "may" or "should" to denote guidance provisions.**
- **Avoid ambiguous provisions, such as "as necessary," "contractor's best practice," "smooth finish," and similar terms.**
- **Use the System Engineering Process to identify requirements. Do not over-specify.**
- **Avoid "tiering." Any mandatory requirement in a document below the first tier, should be stated in the specification.**
- **Only requirement sections of the MIL-STD-491D formats are binding. Do not put requirements in non-binding sections, such as *Scope*, *Documents*, or *Notes*.**
- **Data documentation requirements are specified in a Contract Data Requirements List.**

Figure 5: Guidelines for Writing Specifications [ocw.mit.edu].

Baselines

Baselines explicitly record a product at a certain design definition level. They serve as guides for the progress that will come after. The three traditional baselines functional, allotted, and product are used to create the majority of DoD systems. Even while the program's particular requirements predominate baseline paperwork, they do not alone make up a baseline. The descriptions of the final and enabling products are included in other papers.

Documents defining system requirements, functional architecture, physical architecture, technical drawing package, and requirements traceability often make up end product baselines. A broad variety of papers, such as manufacturing plans and procedures, supportability planning, supply paperwork, manuals, training plans and programmes, test planning, deployment planning, and others, are included in enabling product baseline documents. It is important to assess each enabling product's vulnerability to the effects of system configuration changes. A document should most likely be included as a baseline document if it describes a component of a system and may need to be changed if the configuration changes.

Baselines for Acquisition Programmes

Configuration baselines and acquisition programme baselines are connected. The Programme Baseline must accurately represent the Configuration Baseline's reality, but the two should not be confused. Acquisition Programme Baselines are comprehensive evaluations of the feasibility and maturity of the programme. The system descriptions are called configuration baselines.

III. CONCLUSION

In conclusion, the creation, implementation, and operation of complex systems depend critically on the outcomes of the systems engineering process. These outputs, which include the system requirements, architecture, design, interface specifications, plans for verification and validation, system documentation, and risk management strategies, provide a complete picture of the system and are important resources for stakeholders. The system requirements document, which directs future design and development efforts, encapsulates the intended functions, performance objectives, and restrictions of the system. The system architecture describes the components, subsystems, and relationships between them, as well as the structure and organisation of the system. Stakeholders may better grasp how the system's components work together to provide the necessary functions with the aid of this high-level perspective.

Specifications for the system's parts, connections, and functions are provided through detailed system designs. The system is created in accordance with the specified requirements thanks to the guidance provided by these designs throughout the implementation and integration stages. The communication protocols, physical connections, and data formats required for smooth interaction between system components or subsystems are defined by interface standards. Plans for verification and validation specify the approaches and techniques for assessing the system's overall performance and compliance with requirements. These plans outline the evaluations, testing, and reviews that must be carried out to guarantee that the system functions as planned.

Overall, the results of the systems engineering approach support efficient stakeholder collaboration, communication, and decision-making. They guarantee that the system is built and put into operation in accordance with the stated goals by clearly illustrating the system's requirements, design, and operation. The outputs contribute to the system's overall success by acting as useful resources that direct following tasks such system installation, production, testing, and maintenance.

REFERENCES

[1] D. W. Keith, G. Holmes, D. St. Angelo, and K. Heidel, "A Process for Capturing CO2 from the Atmosphere," Joule, 2018, doi: 10.1016/j.joule.2018.05.006.

[2] A. Lyden, R. Pepper, and P. G. Tuohy, "A modelling tool selection process for planning of community scale energy systems including storage and demand side management," Sustain. Cities Soc., 2018, doi: 10.1016/j.scs.2018.02.003.

[3] L. Abdurrahman, A. Z. R. Langi, and T. M. Simatupang, "Information technology value engineering model and cost efficiency in IT - Based firms," IEEE Syst. J., 2018, doi: 10.1109/JSYST.2017.2663418.

[4] H. Nunns and L. Goentoro, "Signaling pathways as linear transmitters," Elife, 2018, doi: 10.7554/eLife.33617.

[5] X. Dong, X. Du, K. Li, S. Zeng, and B. P. Bledsoe, "Benchmarking sustainability of urban water infrastructure systems in China," J. Clean. Prod., 2018, doi: 10.1016/j.jclepro.2017.09.048.

[6] S. Kannan, T. Sams, J. Maury, and C. T. Workman, "Reconstructing Dynamic Promoter Activity Profiles from Reporter Gene Data," ACS Synth. Biol., 2018, doi: 10.1021/acssynbio.7b00223.

[7] H. M. Beyer et al., "Synthetic Biology Makes Polymer Materials Count," Adv. Mater., 2018, doi: 10.1002/adma.201800472.

[8] J. H. Powell, M. Hammond, A. Chen, and N. Mustafee, "Human Agency in Disaster Planning: A Systems Approach," Risk Anal., 2018, doi: 10.1111/risa.12958.
[9] J. Y. Choi and B. Lee, "Combining LSTM Network Ensemble via Adaptive Weighting for Improved Time Series Forecasting," Math. Probl. Eng., 2018, doi: 10.1155/2018/2470171.
[10] R. Maerani, J. K. Mayaka, and J. C. Jung, "Software verification process and methodology for the development of FPGA-based engineered safety features system," Nucl. Eng. Des., 2018, doi: 10.1016/j.nucengdes.2018.02.009.