OULUN YLIOPISTO

FACULTY OF TECHNOLOGY

EXPLORING THE POTENTIALS AND TOOLS OF SYSTEMS ENGINEERING AND MBSE IN MACHINE DESIGN

Marianne Vanhala

Mechanical Engineering

Bachelors Thesis

August 2023

TIIVISTELMÄ

Systeemitekniikan ja MBSE:n mahdollisuudet ja työkalut koneensuunnittelussa
Marianne Vanhala
Oulun yliopisto, Konetekniikan tutkinto-ohjelma
Kandidaatintyö 2023, 31 s.
Työn ohjaaja(t) yliopistolla: Emil Kurvinen

Tämä opinnäytetyö tutkii Systeemitekniikan (SE) ja Mallipohjaisen Systeemitekniikan (MBSE) käsitteitä modernin koneensuunnittelun kontekstissa. Pääasiallinen tavoite on ymmärtää, miten SE:n monitieteelliset ja kokonaisvaltaiset menetelmät, jotka alun perin juontavat juurensa puhelinalaan, voivat saumattomasti soveltua monimutkaisen koneensuunnittelun maailmaan.

Yksi keskeisistä havainnoista viittaa siihen, että vaikka MBSE herättää kasvavaa kiinnostusta uutena lähestymistapana systeemitekniikkaan, sen tehokkuutta tukevasta konkreettisesta näytöstä on edelleen niukasti saatavilla. Kuitenkin tietyt tutkimukset ovat korostaneet MBSE:n vahvuuksia, erityisesti sen työkalujen kykyä parametriseen ja numeeriseen analyysiin. Nämä työkalut integroituvat saumattomasti suunnitteluprosessin alkuvaiheisiin, mahdollistaen järjestelmän dynaamisen käyttäytymisen jatkuvan tutkimisen.

Vaikka MBSE on edelleen kehittyvä alue, se tarjoaa useita selkeitä etuja, kuten parannetun kommunikaation, lisääntyneen johdonmukaisuuden sekä ajan ja taloudellisten resurssien tehokkaamman hyödyntämisen. Kun otetaan huomioon, että nykyaikainen koneensuunnittelu edellyttää usein yhteistyötä eri alojen erikoisasiantuntijoiden kanssa, voidaan perustellusti väittää, että monimutkaisten koneiden, kuten autojen ja lentokoneiden, suunnittelu kuuluu systeemitekniikan piiriin. Tämän opinnäytetyön tärkein tutkimusmenetelmä oli kirjallisuuskatsaus.

Asiasanat: Systeemitekniikka, MBSE, koneensuunnittelu

ABSTRACT

Exploring the Potentials and Tools of Systems Engineering and MBSE in Machine Design

Marianne Vanhala

The University of Oulu, Degree Program of Mechanical Engineering

Bachelor’s thesis 2023, 31 pp.

Supervisor(s) at the university: Emil Kurvinen

This thesis explores Systems Engineering (SE) and Model-Based Systems Engineering (MBSE) in the context of modern machine design. The primary objective is to understand how SE's interdisciplinary and holistic methodologies, once rooted in the telephone industry, can be seamlessly adapted into the intricate realm of machine design.

One of the key findings suggests that, despite the growing intrigue around MBSE as a novel approach to systems engineering, there is still a lack of concrete evidence to substantiate its effectiveness. However, certain studies have highlighted the strengths of MBSE, especially its tools' capability for parametric and numerical analyses. These tools integrate smoothly with the initial phases of the design process, enabling continuous exploration of a system's dynamic behavior.

While MBSE is still emerging, it offers several apparent advantages, such as improved communication, increased consistency, and efficient use of both time and financial resources. With the knowledge that mechanical engineering these days means working with many different specialists from various fields, we can safely say that engineering machines like cars and planes fall into the realm of systems engineering. The primary methodology employed for data acquisition in this thesis was a literature review.

Keywords: Systems Engineering, MBSE, Machine Design

TABLE OF CONTENTS

TIIVISTELMÄ
ABSTRACT
TABLE OF CONTENTS
1 Introduction 4
2 Systems Engineering 5
  2.1 History 6
3 Systems Engineering Process 8
  3.1 NASE SE-Engine 9
  3.2 Incose Technical Process 11
  3.3 Systems Management 13
    3.3.1 Systems Engineering Management Plan 15
  3.4 Verification and Validation 17
4 Model-Based Systems Engineering 19
  4.1 MBSE Methodologies 22
  4.2 Application of MBSE 25
5 Discussion 27
6 Summary 29
REFERENCES

1 INTRODUCTION

The creation and design of machines and products have become more and more complex, similar to systems all around the world. At this day, a machine cannot anymore be created with only the knowledge of Mechanical Engineers but needs to collaborate with electrical engineers, software engineers, and designers.

The increasing demand for interdisciplinary collaboration calls for tools that can effectively manage these complex structures. This is where Systems Engineering (SE) and, more recently, Model-Based Systems Engineering (MBSE) step in, promising holistic and effective design and realization framework.

In this thesis, I will introduce the concept of systems engineering, the systems engineering process, and the more futuristic tool for more complex systems MBSE. I will discuss the potential of systems engineering and MBSE from a Machine design perspective.

2 SYSTEMS ENGINEERING

Systems Engineering is a holistic approach to creating and leading systems. Its primary objective is to minimize undesirable consequences and ensure that the systems meet the intended goals. It emphasizes the early definition and documentation of stakeholder needs and requirements and effectively captures their functional aspects. Systems engineering is rooted in a system-thinking mindset and therefore understands the system as a complex entity with interconnected components.

In this context, a 'system' is defined as a construct of individual elements producing results that are unattainable by elements alone. These elements can be a wide range of components, including tangible entities such as products, processes, techniques, facilities, and services, as well as intangibles such as people and information. Ultimately, a system is a whole, where interacting parts are organized to accomplish one or more stated purposes.

Systems engineering involves an iterative process of top-down synthesis, development, and operation of real-world systems. It focuses on the design and application of the entire system, considering all the facts and variables involved. Systems engineering uses multiple disciplines to contribute specialized expertise, knowledge, insights, and technology.

While Systems engineering has its roots in the 1930s, its importance has grown exponentially in the context of contemporary complex systems covering multiple facilities and services. These sub-elements are often systems themselves and therefore create complications with autonomies, testing, capabilities, and requirements (INCOSE, 2015, pp. 9-10). This kind of system, which is constructed by other systems, is called a System of Systems (SoS).

2.1 History

The concept of Systems Engineering is relatively modern, but the systems themselves have been engineered since ancient times, including the Roman Aqueducts, Egyptian pyramids, and later the Great Wall of China. However, until the early 1900s, the engineering of these systems was carried out by experienced specialists who applied knowledge gained through their extensive careers. According to Parnell in 2010, the impetus for more disciplined systems engineering came with the boom of the telephone industry in the 1920s and 1930s. Consequently, the term Systems Engineering was defined by Bell Telephone Laboratories in the early 1940s (INCOSE, 2023).

During World War II, systems thinking and mathematical modeling became integral to enhancing military operations, and even after the war systems engineering was widely used in complex weapon systems (Parnell, 2010). From the Systems Engineering Body of Knowledge (SEBoK), we can learn that by 1969, systems engineering had reached the United States Department of Defense and they released their military standard for Systems Engineering Management (Military Standard 499). Later they updated their guidance (MIL-STD-499A). These standards covered the process of systems engineering and gave instructions for the Systems Engineering Management Plan (SEMP) and for the task statements.

Furthermore, in the 1960s systems engineering pioneer Arthur Hall was one of the first to start teaching systems engineering (SEBoK, 2023). From the Hagley Library collection (2023) we can learn that in 1967, Hall accepted a position as a visiting professor of systems engineering at the University of Pennsylvania. While there, he created the first Ph.D. program for systems engineering in the United States. Hall supported his teachings with a book called "A Methodology for Systems Engineering". In the book Hall identified 5 traits of an ideal systems engineer (SEBok, 2023):

1. Understanding the Systems
2. Faculty of Judgement
3. Creativity

4. Facility in Human Relations
5. Facility for Expression

In 1989 the National Council on Systems Engineering (NCOSE) was founded when it was noted that there was a need for more formally trained systems engineers. Later on, this council would change its name to the International Council on Systems Engineering, INCOSE, with the lead of Dr. Brian Mar (SEBok, 2023). In the coming decade, INCOSE would establish the journal "Systems Engineering" and later INCOSE's Systems Engineering Handbook.

INCOSE wasn't the only organization interested in developing systems engineering. Before INCOSE published its handbook, the National Aeronautics and Space Administration (NASA) had already made public its own Handbook (NASASP-6105) standardizing the fundamental concepts and techniques inside NASA's personnel.

Later on, in the 2000s the International Standards Organization created standards for systems engineering application and management. The standards defined the entire systems engineering lifecycle with customer needs, requirements, and constraints. These standards later developed into the current ISO 24748/15288/12207 (SEBok, 2023).

3 SYSTEMS ENGINEERING PROCESS

The Systems engineering process has different variations adapted to specific contexts. In this thesis, our attention is directed towards two prominent systems engineering approaches: NASA's SE Engine (NASA, 2007) and the ISO/IEC/IEEE 15288 (2023) standard, which was collaboratively developed with INCOSE. Despite their distinct origins, both processes share fundamental similarities, as shown in Figure 1, where their process timelines are compared with each other.

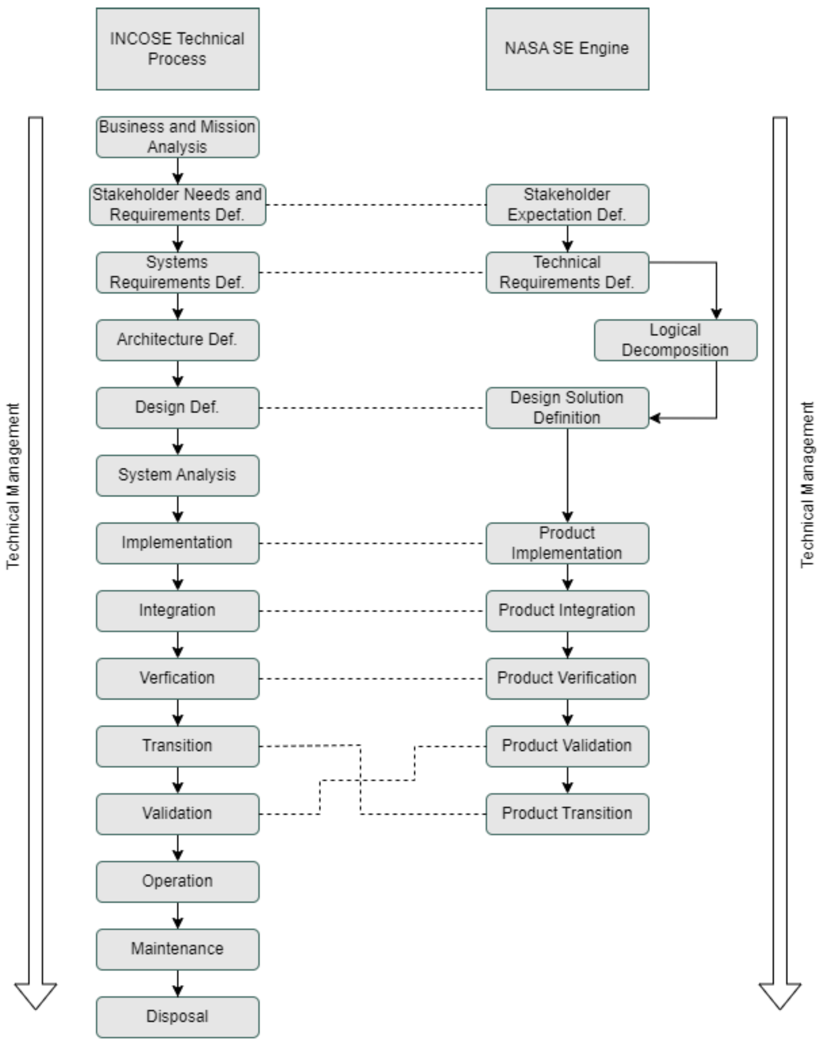

Figure 1. Comparison between INCOSE Technical Process and NASA SE Engine

3.1 NASE SE-Engine

SE Engine is introduced in the NASA Systems Engineering Handbook (2007). It is created with three distinct technical processes, each of which encompasses a set of activities. These processes are derived from NASA's Procedural Requirements NPR 7123.1, *NASA Systems Engineering Processes and Requirements.* The three processes are the System Design Process, the Product Realization Process, and the Technical Management.

The flow of the SE Engine is structured in such a way that the System Design Process proceeds in a top-down manner, following the product's different layers from the whole product to its smallest element. Subsequently, the Product Realization Process moves through the layers of the product in a bottom-up fashion. This sequential process is illustrated in Figure 2. Simultaneously, Technical Management plays an important role across the entire procedure, in a cross-cutting manner. An exploration of the Technical Management process will be presented in Chapter 3.3.

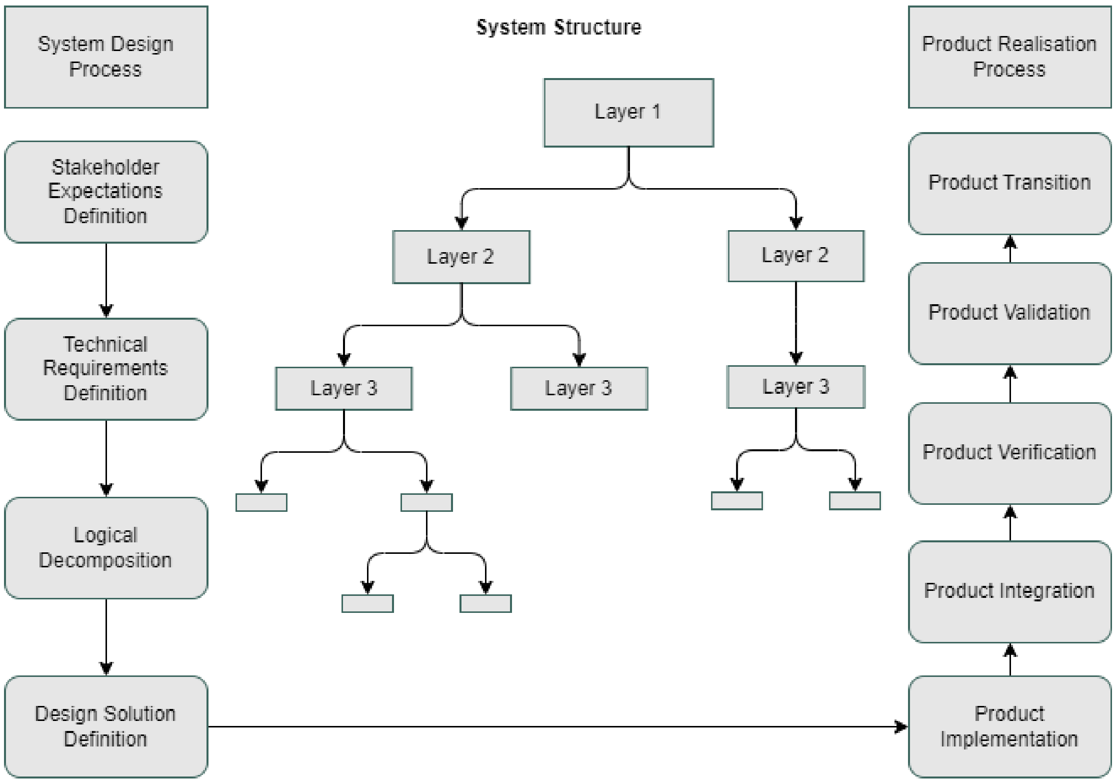

Figure 2. NASA's SE Engine

In NASA's Procedural Requirements NPR 7123.1, SE Engine Processes are divided into multiple activities.

System Design Process

Stakeholder Expectations Definition: In this activity, the goal is to identify and gather the stakeholder expectations to establish the product's features and other requirements. Additionally, these expectations contribute to the definition of the Measures of Effectiveness (MOEs).

Technical Requirements Definition: During this activity, the initial expectations are transformed into quantifiable and measurable technical requirements. In NASA, these requirements are unified using "shall" statements. This phase also introduces the concept of Measures of Performance (MOPs) and Technical Performance Measures (TPMs).

Logical Decomposition: The goal of this activity is to enhance comprehension of the technical requirements and their interrelationships. It involves converting the requirements into a series of logical decomposition models.

Design Solution Definition: Building upon the outcomes of Logical Decomposition, this phase reinterprets them into concrete design solutions. Multiple alternatives are generated, from which the most suitable is selected. The chosen solution is then refined into a final design definition that aligns with the requirements.

Product Realization Process

Product Implementation: During this stage, the specified product is brought to life through acquisition, manufacturing, or reuse.

Product Integration: The previously implemented lower-level components are brought together to form a higher-level end product.

Product Verification: The final product's ability to fulfill the specified requirements specified in the Design Solution Definition is demonstrated in this activity.

Product Validation: This phase confirms that the verified end-product performs as intended in its operational context. The focus is placed on identifying and resolving any anomalies before the product delivery, particularly addressing the designated MOEs.

Product Transition: As the last activity on SE Engine, the verified and validated product is handed over to the customer or intended end user.

3.2 Incose Technical Process

The INCOSE Systems Engineering Handbook (2015) introduces the standard ISO/IEC/IEEE 15288:2015. It’s noteworthy that this standard has later been updated to ISO/IEC/IEEE 15288:2023. To ensure the information is not outdated the newer version has been reviewed. ISO/IEC/IEEE 15288 divides the systems engineering process into 14 distinct segments: Business and mission analysis, stakeholder needs and requirements definition, system requirements definition, architecture definition, design definition, system analysis, implementation, integration, verification, transition, validation, operation, maintenance, and disposal. This list can be additionally revised from Figure 1. Upon examining the list, we can see that standard ISO/IEC/IEEE 15288 addresses the entire life cycle of the system, from identifying the opportunities to the point when the system is no longer needed.

During the technical process, maintaining traceability of analysis results, decisions, and the foundational information for those decisions is crucial. This information must be stored in a manner that facilitates accessibility and modification throughout the system’s life cycle. To address this need, INCOSE introduces the Requirements Verification and Traceability Matrix (RVTM).

Strategy bridge (2023) introduces RVTM as:

“A document showing the source and parent/child relationships among requirements and how they cascade from the system level to the lowest level of the system architecture. The verification method, result, and resolution for each requirement are included “.

In the process introduced in ISO/IEC/IEEE 15288, RVTM is initially created during Business and Mission Analysis and finalized during the verification process. During the whole process, it should be updated with up-to-date information.

Before every step in the process, it is important to prepare. This means collecting the necessary information, creating a plan on how to go through the steps, and ensuring that necessary resources are available for use when needed.

Business and Mission Analysis: Defines business or mission problems or identifies an opportunity. It then characterizes the solution space and determines possible solution classes. Creates the initial RVTM (Requirements verification and Traceability matrix).

Stakeholder Needs and Requirements Definition: Defines the stakeholder’s requirements. Defines stakeholders who will participate with SE in the system's life cycle.

System Requirements Definition: Transforms the requirements into a more technical sense. Additionally identifies the stakeholders who impose unavoidable restrictions on the system.

Architecture Definition: Generates alternative architecture designs that fulfill the requirements listed. Prioritizes the requirements and identifies functions and constraints that have significant effects on the architecture decisions.

Design Definition: Provides enough data and information that are detailed enough to enable implementation. Identifies needed technologies and establishes design characteristics and design enablers related to each system element.

System Analysis: Provides data for technical understanding to help make decisions throughout the life cycle. Plans and performs the analysis.

Implementation: Fabricates, codes, or builds each system element. Additionally develops and prepares the training data for future users. At this step, we have the complete specifications.

Integration: Collects all the implemented elements and synthesizes them into a realized system. This system is assembled in a way that fulfills the requirements of stakeholders.

Verification: Provides objective evidence that the system is working and fulfilling its requirements.

Transition: Confirms the system, in its operational environment, is capable of providing services that fulfill the stakeholder requirements. Involves creating a strategy that contains operator training, delivery strategy, and problem resolution strategy.

Validation: Provides objective evidence that the system, when in use in its operational environment, fulfills the stakeholder requirements and achieves its intended use.

Operation: Uses the system to deliver its services and supports the customer.

Maintenance: Sustains the capability of the system to deliver its services. This includes logistics support.

Disposal: Ends the existence of a system and disposes of it correctly while identifying the possible reuse possibilities of different parts of the system.

3.3 Systems Management

Systems engineers are continuously interacting with project management but their point of view of project lifecycle is often defined differently. Project managers define the project lifecycle from the start of the project to the end of the project, contrarily systems engineers think it as a product idea to product disposal. The ISO/IEC/IEEE 15288 standard defines the technical management processes as a way to establish, evolve, and execute plans and a way to assess achievements and progress.

Systems Management usually starts with the creation of plans for the system creation project, which includes creating a Systems Engineering Management Plan (SEMP). All the necessary information about the project is written into it, like the roles and responsibilities of different stakeholders, the project schedule, and assessment dates.

Assessments are an important part of systems management to be able to define the current circumstances around the project, the plans followed, and whether the project is following the schedule. With the assessments the project plans are aligned and checked if they are feasible, later on, assessment helps to define the project status and assess the technical and process performance (INCOSE, 2015).

Part of systems management is risk management, which includes both safety of the personnel and technical risks. Identifying, analyzing, treating, and monitoring the risks that happen throughout the whole project continually (INCOSE, 2015). ISO/IEC/IEEE 16085 standard gives instructions on how to properly manage the risks and standardizes the common terminology.

Both NASA and INCOSE have their processes for systems management but they have a lot of similarities with each other. This can be seen in Figure 3, where the processes are listed next to each other. These processes are introduced in more detail in NASA’s Systems Engineering Handbook and INCOSE’s Systems Engineering Handbook.

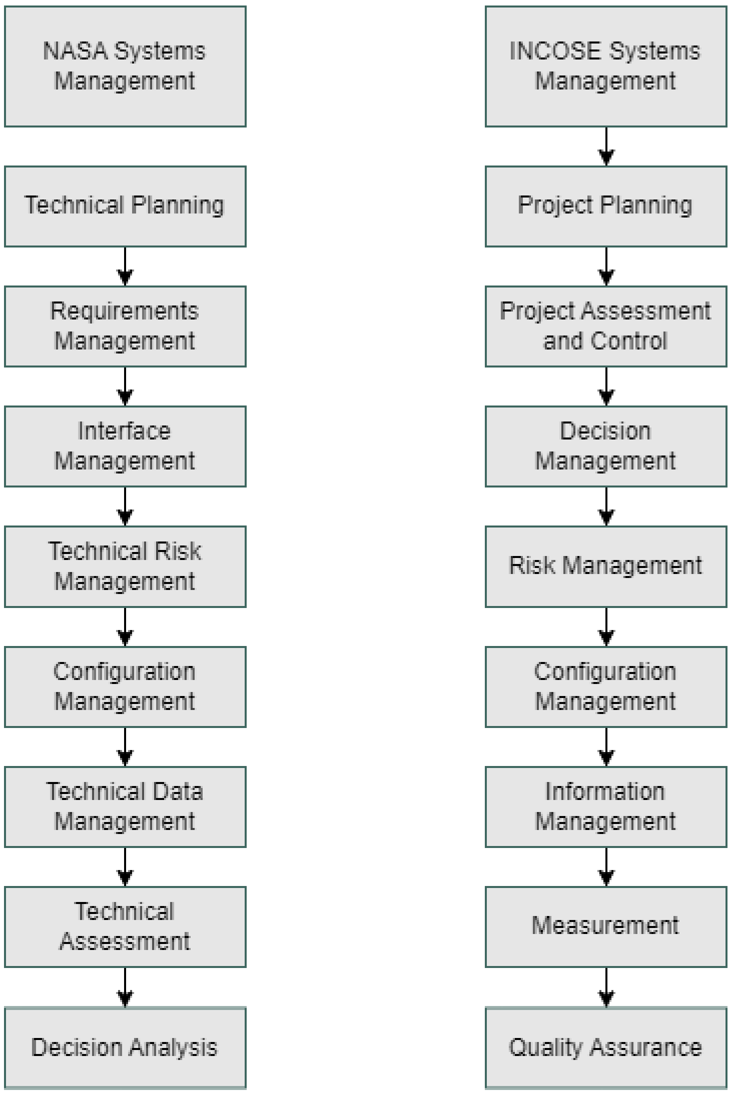

Figure 3. NASA's and INCOSE's Systems Management Processes

3.3.1 Systems Engineering Management Plan

A Systems Engineering Management Plan (SEMP) was created by Tamara S. Rodriquez (2009) to support the management of systems throughout the systems' lifecycles. It gives technical instructions on all activities throughout the effort of systems engineering, but instead of only giving advice for activities like collecting stakeholder needs or checking the viability of the parts, it instructs on assignment responsibilities, formal and informal reviews, and risk management.

SEMP should be created immediately when the project is launched and it should be available for all stakeholders. A comprehensive SEMP outline should include at least the three core areas:

1. Technical Project Planning and Control: This core area should contain the details of the project like the scope, roles and responsibilities, schedule, project plan, and reviews. It also should include the management strategies and protocols for the project itself and issues that might arise during it. Additionally, metrics for success measurements and Verification and Validation procedures should be defined.
2. Systems Engineering Process: This section should contain details for the processes guiding activities during each phase and a definition of the entire systems lifecycle from initial concept to disposal.
3. Engineering Specialty Integration: This phase contains the explanations of which engineering specialties are included in the project management. These specialties are for example Human engineering, safety protocols, and quality assurance.

This outline is also visualized in Figure 4.

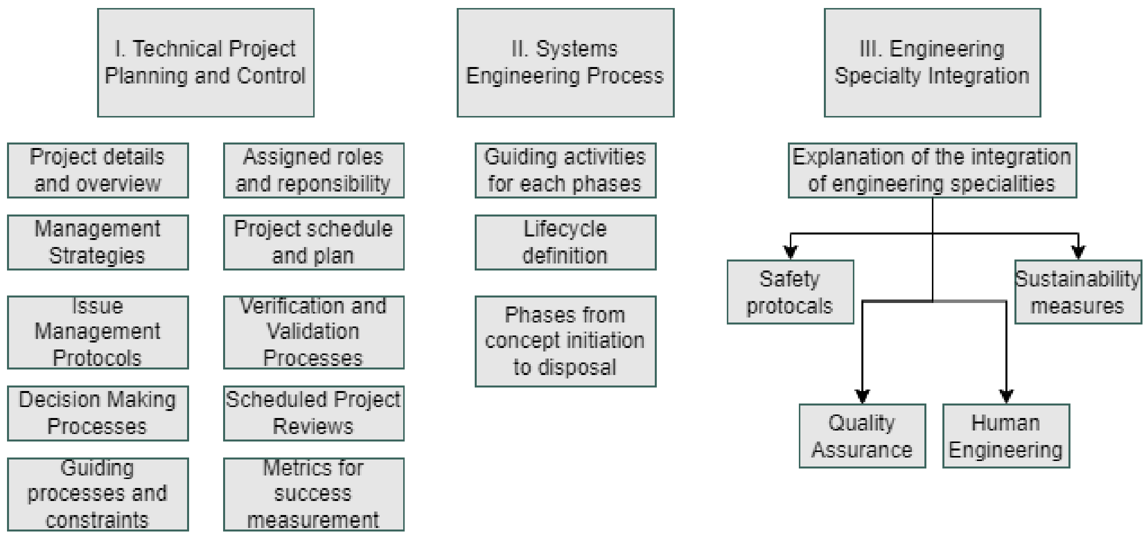
I. Technical Project Planning and Control
Project details and overview
Assigned roles and reponsibility
Management Strategies
Project schedule and plan
Issue Management Protocols
Verification and Validation Processes
Decision Making Processes
Scheduled Project Reviews
Guiding processes and constraints
Metrics for success measurement
II. Systems Engineering Process
Guiding activities for each phases
Lifecycle definition
Phases from concept initiation to disposal
III. Engineering Specialty Integration
Explanation of the integration of engineering specialities
Safety protocals
Sustainability measures
Quality Assurance
Human Engineering

Figure 4. SEMP outline

SEMP should be updated whenever it's necessary and finalized at the end of the project.

The benefit of using SEMP is that it’s a comprehensive approach to systems management. SEMP includes the project plan but it also addresses how the project plan would be managed and how the resources would interact with each other through the lifecycle activities. Additionally, SEMP helps to manage the system as a whole instead of only the technical aspects. In addition to the last two, SEMP can be used as a road map, it would guide, communicate, and document the project of system development and management. And lastly, it enables success through comprehensive planning, reduction of rework due to lack of processes and miscommunication, efficient management, and collaboration. (Rodriquez, 2009)

3.4 Verification and Validation

Verification and Validation (V&V) is a process during which it is confirmed that the realized product meets all its specifications and design descriptions. Verification testing answers the question if the system is done right and the validation process answers if the right system was realized (NASA 2007, p.83). Both INCOSE and NASA place pressure on having a proper V&V process.

In Verification testing, which is the first of the two performed, the product is tested against the approved requirements set (NASA 2007, p.83). The final purpose of verification is to get objective evidence that the requirements and characteristics are fulfilled (INCOSE 2015, p. 83). Different types of verification include analysis, demonstration, inspection, and tests. Some testing includes the ones that are used to assist the development, or manufacturing of the product, or any engineering-type test to define the technical progress and to verify design risks are minimized (NASA 2007, p.83).

Validation again is a process where it is determined product’s effectiveness in fulfilling its duties and suitability for the job. Stakeholder expectations should be fulfilled and any anomalies discovered and resolved before the final transition to end users (NASA 2007, p.98). A Requirements and Validation Traceability Matrix can be used to track the data while performing the validation process. This ensures the selected operational scenarios are validated.

4 MODEL-BASED SYSTEMS ENGINEERING

Systems have been growing more and more complex in the last decades. This created a demand for a tool to manage these complex systems and Model-Based Systems Engineering (MBSE) has been a choice for many systems engineers. MBSE is a relatively new approach but at the moment it is one of the major research themes in the systems engineering field and was chosen as INCOSE's primary component for the SE vision 2025 (INCOSE, 2023).

The motivation to change to MBSE from the more traditional document-centric approach has been arising from the older approaches’ shortcomings. With the ever-growing size and complexity of the systems traditionally-used Document-Centric Systems Engineering (DCSE) methods are less effective. Documentation tends to gradually become more incomplete and inconsistent, and the risk of overlooking critical information rises when the systems are more complex and the amount of system requirements grow (Rogers III and Mitchell, 2021).

MBSE is built around a model and this model is the center of all the system development activities. This means all information is stored and managed in a central repository. With this method, complex systems can be managed while maintaining consistency and traceability. Different parts in the model are strongly linked with each other, enabling comprehensive traceability while maintaining an audit trail for all data generated, decisions made, and information supplied. Different documentation reflecting the state of the system can be automatically developed for different stakeholders. As a result of this, the unnecessary time engineers spend on document searching and assembling reports can be used more effectively (Rogers III and Mitchell, 2021).

Often the architecture-defining process is rushed through, even with the growing awareness of the importance of rigorous architecture. This disregards the fact that the greatest freedom and the largest number of solutions exist when the problem is first being defined. Consequently, this rushing affects the design and later on, brings challenges during the integration process (Rogers III and Mitchell, 2021).

With MBSE, the generation of architectures starts from the simplest components and builds it ultimately into a system from the realized components. Often, they facilitate the reuse of component models within the architecture. During this process, the information is stored in a way that it can be accessed and modified if needed.

Other deficiencies can also arise with the use of the traditional DCSE method. Table 1 introduces the deficiencies already mentioned and some others in the current method of system development and possible solutions MBSE offers.

Table 1. Deficiencies of current methods and MBSE Solutions

DEFICIENCY	**POSSIBLE SOLUTION WITH MBSE**
Wrong features	Interconnectivity enables the automatic propagation of error identification.
Rushing architecture	A meticulous bottom-up approach to architecture development
Wasted time searching for information and assembling reports	Different documents can be automatically extracted from the system
Incomplete and/or inconsistent documentation	Information is stored in a central repository and all linked data will be automatically updated
Overlooking critical information	Information found in the system is always up-to-date.
Unstandardized vocabulary between stakeholders	The MBSE model can be tailored to fit specific stakeholders and their language

A case study from Madni and Sievers (2018) shows that significant improvements in quality and cost can be achieved with MBSE. Other benefits of changing to MBSE included centralized documentation, managing increased system complexity and commonality, automated data validation, and enhanced traceability between capability, requirement, function, and deployment.

Rogers III and Mitchell (2021) introduced us to more benefits. Most of the model-based approaches today are easy to use and can be tailored to fit many different kinds of circumstances and backgrounds. Based on the model and language used, some features can be completely ignored or added later if they are needed. This created a possibility for different stakeholders to examine the model from their perspective without unnecessary functions or features.

Many other papers also discuss the potential benefits and advantages of MBSE. A study conducted by Henderson and Salado (2020) shows the 12 benefits that were at least 1 % of the total number of benefits.

1. Better communication/information sharing.
2. Increased consistency
3. Reduce cost
4. Reduce time
5. Reduce errors
6. Improved system understanding
7. Increased productivity
8. Better analysis capability
9. Increased efficiency
10. Early verification and/or validation
11. Reduce ambiguity
12. Higher level support for automation

The Spread of the benefits mentioned across all the papers was wide. The same study suggests that affirm that there is no commonly accepted evidence or benefits. They were able to find that most of the papers discussing MBSE, didn't have measured proof of their claim of these advantages. Mostly the advantages were perceived, or they were referencing another paper. They still wanted to point out that the absence of empirical evidence doesn't imply that MBSE is disadvantageous. Often perceived information is useful and valuable from the practitioner's standpoint.

4.1 MBSE Methodologies

Currently, there are two distinct modeling approaches available in the market. These approaches can be categorized into those that support graphical representations and those that incorporate semantic representation in addition to graphical (Rogers III and Mitchell, 2021).

Some of the known approaches that support graphical representations are Unified Modeling Language (UML), Systems Modeling Language (SysML), and Object Process Methodology (OPM).

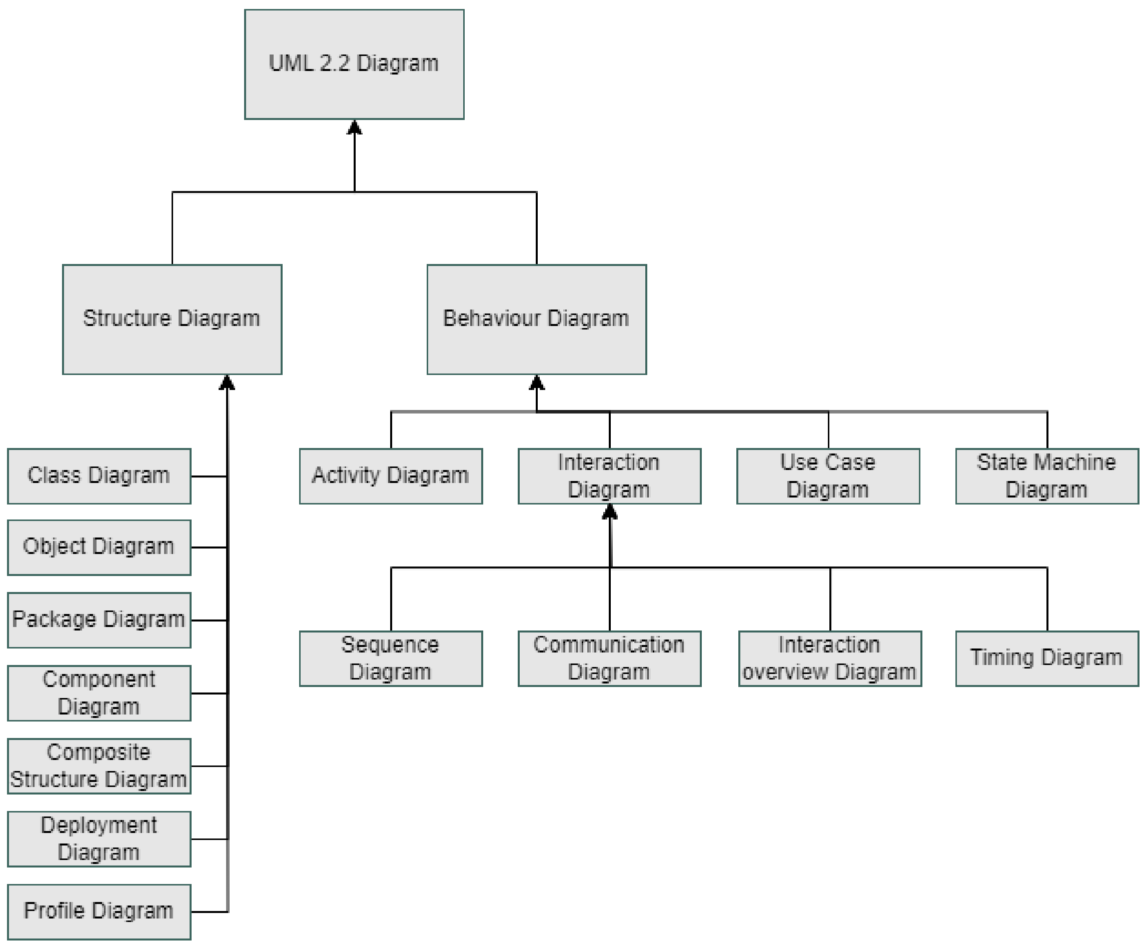

Figure 5. UML 2.2 Diagrams Structure (UML, 2011)

UML is a standardized modeling language made for software systems but it can be used in business and non-software systems modeling too (UML, 2023). UML supports visualizing and documenting models in a way that meets all the requirements. The visualization often happens with diagrams, which, in UML, can be divided into 3 types: behavior diagrams, interaction diagrams, and structure diagrams (Brooch et al., 2005). In the newest version of UML provided by Object Management Group (OMG), interaction diagrams have been categorized under behavior diagrams (UML, 2011) (Figure 5).

SysML extends UML for more general systems modeling. It contains all commonly used design methodologies like “V”, Spiral, and Waterfall. These methodologies are visualized in Figure 6.

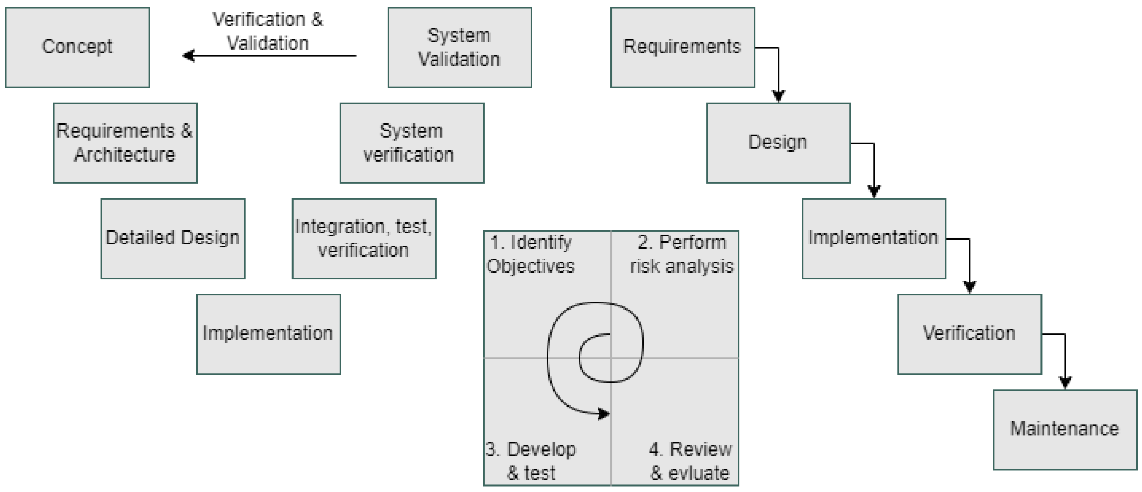

Figure 6. Known design methodologies left to right: "V", Spiral, and Waterfall

Both the UML and SysML are flexible in how they are used. This quality has some criticism because the flexibility of models can contribute to ambiguity. Sometimes the way diagrams or models operate may not be clear or may depend on the tool used to interpret the model (Madni and Sievers, 2018).

The last approach that supports only graphical modeling is Object Process Methodology (OPM), which comprises a small set of building blocks consisting of objects and links.

Structural links show how objects relate to each other and are represented in the Object Process Diagram (OPD) which is the only diagram in OPM, compared to UML which has 14 different kinds of diagrams. Each of these OPDs is represented in Object Process Language, which is a subset of natural English. This makes the language interpretable by humans and by computers (Dori 2016).

Approaches that support both graphical and semantic approaches are for example Foundational UML (fUML), Ptolemy, and State Analysis. Each of these has on top of the computation models, a human interpretable graphical representation (Madni and Sievers, 2018).

fUML, along with its Action Language for fUML (ALF), is a part of UML that can be directly executed. It’s a precise and limited version of UML that maintains communication benefits but restricts what can be shown and how models are made. fUML has three main components: a well-defined and complete subset of UML, a way to execute the models based on fUML’s rules, and a basic library of essential components.

Ptolemy is a modeling tool designed for precise modeling of complex systems. It uses an approach called super-dense time”, where time is divided into small steps within a continuous timeline. Ptolemy’s structure is hierarchical in that each subset has its clear meaning and can be analyzed independently. One of Ptolemy’s key strengths is its ability to handle diverse synchronization methods and complex systems by breaking them into manageable submodels.

State Analysis is a more formal process for modeling control systems, defining boundaries, and specifying relationships. The model’s semantics, encompassing elements like state variables, controllers, and commands, offer a precise blueprint for software development and verification. The process begins with modeling the controlled system’s behavior and guiding system functionality development.

4.2 Application of MBSE

The previously mentioned case study from Rogers III and Mitchell (2021) started its change from DCSE to MBSE without much solid evidence that MBSE would bring a positive Return of Investment ROI. Integration of MBSE into the Submarine Warfare Federated Tactical Systems (SWFTS) started after they started to notice how their document-based system was lacking behind the quality, they expected from it. They started their transition in 2010 and lasted for 2 years.

The results they were able to get were game-changing. In some parts of their system, they were able to reduce the cost by 18 % and were able to detect the defects 18% better. This enabled them to remove the faulty delivering higher-quality products to customers. In the initial 5 years, they saved an estimated $3.65 million to $25.6 million, indicating an ROI of 1.1 to 7.75.

Other benefits they were able to record were a 9% reduction in overall interface defects, centralized documentation, better management of complex systems, automated data validation, and improved traceability between capability, function, deployment, and requirement.

Akundi and Lopez (2021) discuss the application of MBSE in different activities in industry and system life-cycle.

Industries embracing Industry 4.0, characterized by increased automation, can leverage MBSE methodologies. Effective communication between the stakeholders at the system's inception optimizes time and resources by minimizing iterative loops (A17). The use of MBSE language, such as SysML, facilitates collaboration among stakeholders, leading to optimal solutions.

MBSE is a solution to make industries change into smart manufacturing systems, but unfortunately, there is still a lack of willingness to transition.

MBSE can be beneficial for lifecycle design and management activities. Systems Lifecycle Management (SLIM) is a combination of MBSE and project lifecycle management that increases the usage of modeling language across all stages in the systems development lifecycle (Bajaj *et al.*, 2011). MBSE's focus on early system development aspects creates a continuity gap. This gap could be covered by SLIM on top of being a holistic model that spans different system aspects.

Akundi and Lopez (2021) refer to two papers published by Hummel et al. (2015) and Bretz et al. (2016) on more benefits the usage of MBSE would bring to manufacturing processes. It would make it more feasible to optimize ROI, easier customizability of products to manufacturers, significantly easier access to information, and interconnection between different domains in engineering and process activities. They also point out that it is of the utmost importance for the stakeholders to understand every phase in the systems development lifecycle and this is what MBSE languages and tools enable.

The adoption of MBSE in larger enterprises faces challenges due to their less methodical approach and resistance to change. In contrast, small and medium-sized enterprises could highly benefit from the implementation of MBSE, because it has shown to increase efficiency and productivity.

5 DISCUSSION

When we think about the design and realization process of machines and products there are a lot of parts that the mechanical engineer cannot touch. The creation of any machine be it a phone, coffee machine, car, or space rocket, is interdisciplinary. A phone needs engineers from the fields of software, electronics, mechanical, and computer, and designers from different backgrounds from branding and style to machine and electrical. Similarly, a car is not only made of mechanical parts but also electronics and software.

Systems engineering and Model-Based Systems Engineering have been made to support these complex interdisciplinary structures. Mechanical engineering and therefore machine designing is then one part of systems engineering and it can benefit from all the tools of systems engineering.

Tools that are especially beneficial in machine designing are mostly from MBSE. Being able to create bonds between different machine elements and assign requirements for each becomes easier with the highly interlinked items inside models. The requirements are automatically updated in all linked elements and necessary documents can be automatically uploaded.

The machine designer needs to be able to understand what is being done in the other parts of the machine design project. This way they can accommodate the necessary space needed for different electrical or design elements. Communication is the key to this and MBSE facilitates effective communication among diverse stakeholders. Effective communication minimizes the iterative loops which optimizes the time and resource usage.

MBSE is one tool to modernize the machine design process but it still needs work to be able to be most efficient. A lot of research on MBSE has been done in the last few years but many industrial companies are still reluctant to change to MBSE from the more document-based approach in systems engineering but this is quite normal in the field of industry. Change seems to be hard.

In research made in 2016, Eugenio Brusa and Davide Ferretto tested the impact of MBSE in designing a mechatronic flywheel-based energy storage system. The most noticeable strength of MBSE is the tools to conduct parametric and numerical analyses while integrating seamlessly with the earlier phases of the design process. This enables a continuous exploration of the dynamic behavior of the system.

Because testing and evaluating is one of the main parts of the machine design process MBSE can based on Brusa's and Ferretto's research be streamlined and made more effective.

While I think MBSE still needs to be somewhat modified to seamlessly fit into machine design, it has the potential to revolutionize how we approach and execute machine design processes. In the coming future, I expect the creation of many computational tools to help the work of machine designers. This could be an AI tool to create CADs from old technical drawings or give solution options for different identified problems.

6 SUMMARY

In the 21st century, modern machine design is faced with the challenge of ever-increasing system complexity, demanding a collaborative approach across various engineering disciplines. While traditional methods like document-based approaches still work, they are becoming less adaptive to the more intricate demands of modern projects.

For the last half a century systems engineering has offered a structured way to handle complexities of the interdisciplinary projects. This is accentuated by findings from Brusa’s and Ferretto’s research, highlighting MBSE's capability for seamless integration with diverse design phases, markedly enhancing testing and evaluation processes.

Despite the existence of more modern tools, MBSE faces resistance in industrial adoption, largely due to the natural resistance to change.

Nevertheless, with INCOSE's Systems Engineering Vision 2025 spotlighting MBSE, there's renewed optimism. While there's an acknowledgment that MBSE needs further refinement, its forward-thinking approach signifies the potential for a transformative era of more integrated and efficient machine design.

REFERENCES

Akundi, A. and Lopez, V. (2021) ‘A Review on Application of Model-Based Systems Engineering to Manufacturing and Production Engineering Systems’, Procedia Computer Science, 185, pp. 101–108. Available at: https://doi.org/10.1016/j.procs.2021.05.011.

Bajaj, M. et al. (2011) ‘4.3.1 Satellites to Supply Chains, Energy to Finance —SLIM for Model-Based Systems Engineering’, INCOSE International Symposium, 21(1), pp. 368–394. Available at: https://doi.org/10.1002/j.2334-5837.2011.tb01214.x.

Booch, G., Rumbaugh, J. and Jacobson, I. (2005) The Unified Modeling Language User Guide. 2nd edition. Addison-Wesley Professional (Addison-Wesley Object Technology Series). p. 495. ISBN 978-0-321-26797-9

BORCHANI, M.F. et al. (2018) ‘Mechatronic System Design using Model-Based Systems Engineering and Set-Based Concurrent Engineering Principles’, in 2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics. 2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, pp. 32–38. Available at: https://doi.org/10.1109/MECATRONICS.2018.8495824.

Brusa, E. and Ferretto, D. (2016) ‘Impact of the MBSE on the design of a mechatronic flywheel-based energy storage system’, in 2016 IEEE International Symposium on Systems Engineering (ISSE). 2016 IEEE International Symposium on Systems Engineering (ISSE), pp. 1–8. Available at: https://doi.org/10.1109/SysEng.2016.7753145.

Dori, D. (2016) Model-Based Systems Engineering with OPM and SysML. 1st ed. 2016 edition. New York: Springer. p. 411. ISBN 978-1-4939-3294-8

Hagley Museum and Library Archives (2023) Collection: Arthur D. Hall III papers. Available at: https://findingaids.hagley.org/repositories/3/resources/1171 (Accessed: 9 August 2023).

Henderson, K. and Salado, A. (2021) ‘Value and benefits of model-based systems engineering (MBSE): Evidence from the literature’, Systems Engineering, 24(1), pp. 51–66. Available at: https://doi.org/10.1002/sys.21566.

INCOSE (2000) History of Systems Engineering. Available at: https://www.incose.org/about-systems-engineering/history-of-systems-engineering (Accessed: 13 June 2023).

INCOSE (2015) INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. 4th edition. Hoboken, New Jersey: Wiley. p. 304. ISBN 978-1-1189-9940-0

INCOSE (2023) Systems Engineering Vision 2035. Available at: https://violin-strawberry-9kms.squarespace.com (Accessed: 9 August 2023).

ISO/IEC/IEEE 15288:2023 (2023) Systems and software engineering — System life cycle processes. International Organization for Standardization ISO: p. 116. Available at: https://www.iso.org/standard/81702.html (Accessed: 13 June 2023).

ISO/IEC/IEEE 16085:2021 (2021) Systems and software engineering — Life cycle processes — Risk management. International Organization for Standardization ISO: p. 47. Available at: https://www.iso.org/standard/74371.html (Accessed: 22 August 2023).

ISO/IEC/IEEE 24748-1:2018 (2018) Systems and software engineering — Life cycle management — Part 1: Guidelines for life cycle management. International Organization for Standardization ISO: p. 72. Available at: https://www.iso.org/standard/72896.html (Accessed: 22 August 2023).

ISO/IEC/IEEE 24748-4:2016 (2016) Systems and software engineering — Life cycle management — Part 4: Systems engineering planning. International Organization for Standardization ISO: p. 62. Available at: https://www.iso.org/standard/56887.html (Accessed: 13 June 2023).

Madni, A.M. and Sievers, M. (2018) ‘Model-based systems engineering: Motivation, current status, and research opportunities’, Systems Engineering, 21(3), pp. 172–190. Available at: https://doi.org/10.1002/sys.21438.

National Aeronautics and Space Administration. (2007) NASA Systems Engineering Handbook. Washington, D.C.: NASA Headquarters. p. 356. ISBN 978-1-5372-7668-7

Object Management Group (2011), OMG Unified Modeling Language™ (OMG UML) Superstructure Version 2.4.1. Available at: https://www.omg.org/spec/UML/2.4.1/Superstructure/PDF

Parnell, G.S. (2010) ‘Introduction to Systems Engineering’, in Decision Making in Systems Engineering and Management. John Wiley & Sons, Ltd, pp. 183–195. Available at: https://doi.org/10.1002/9780470926963.ch6.

Rodriguez, T.S. (2009) Systems engineering management plans. SAND2009-7836. Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States). Available at: https://doi.org/10.2172/1004370.

Rogers III, E.B., and Mitchell, S.W. (2021) ‘MBSE delivers significant return on investment in evolutionary development of complex SoS’, Systems Engineering, 24(6), pp. 385–408. Available at: https://doi.org/10.1002/sys.21592.

SEBok (2023) Brief History of Systems Engineering. Available at: https://sebokwiki.org/wiki/Brief_History_of_Systems_Engineering (Accessed: 13 June 2023).

Strategy Bridge International (2023) Systems Engineering Training, Systems Engineering Glossary, Requirements Verification Traceability Matrix (RVTM). Available at: https://www.systemsengineeringtraining.com/systems-engineering-glossary/requirements-verification-traceability-matrix (Accessed: 5 July 2023).

Unified Modeling Language (2023) What is UML | Unified Modeling Language (no date). Available at: https://www.uml.org/what-is-uml.htm (Accessed: 12 August 2023).