Informatics

Navigating the MBSE Transition: A Meta-Synthesis

Herausforderungen, Tücken und bewährte Praktiken bei der Einführung in der Praxis

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Tobias Henöckl, BSc
Matrikelnummer 11709476

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Wirtsch.Inf.Univ. Dr.rer.pol. Dominik Bork

Wien, 26. November 2024

Tobias Henöckl

Dominik Bork

Informatics

Navigating the MBSE Transition: A Meta-Synthesis

Challenges, Pitfalls, and Best Practices in Real-World Adoption

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Tobias Henöckl, BSc
Registration Number 11709476

to the Faculty of Informatics

at the TU Wien

Advisor: Associate Prof. Dipl.-Wirtsch.Inf.Univ. Dr.rer.pol. Dominik Bork

Vienna, November 26, 2024

Tobias Henöckl

Dominik Bork

Erklärung zur Verfassung der Arbeit

Tobias Henöckl, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 26. November 2024

Tobias Henöckl

Danksagung

Ich möchte meinem Betreuer, Dominik Bork, meine Anerkennung für seine Anleitung und sein Feedback während der Entwicklung dieser Arbeit aussprechen. Sein Beitrag war sehr wertvoll für die Verfeinerung und Gestaltung des Endergebnisses.

Ich bin auch meiner Familie und meinen Freunden dankbar, die mich während des Schreibprozesses unterstützt und mit mir diskutiert haben, insbesondere was die Struktur und Organisation der Arbeit betrifft. Ihr Zuspruch hat mir in dieser intensiven Arbeitsphase den dringend benötigten Ausgleich verschafft.

Diese Arbeit basiert auf dem Ansatz einer Meta-Synthese, der sich stark auf die Arbeit zahlreicher Forschenden vor mir stützt. Ich bin sehr dankbar für das kollektive Wissen und die Einsichten der akademischen Gemeinschaft, die diese Studie ermöglicht haben. Ihre Beiträge waren für das Verständnis der Komplexität des Themas und für die Ausarbeitung dieser Studie von wesentlicher Bedeutung.

Bei der Entwicklung und aufwändigen Fertigstellung dieser Arbeit war ich dankbar für eine Reihe von Tools, Methoden und Ressourcen, die den Arbeitsfortschritt unterstützt haben. Diese Arbeit ist ein Produkt kollektiven Wissens und kollegialer Zusammenarbeit, und ich freue mich, meinen Teil zu diesem Thema beitragen zu können.

Acknowledgements

I would like to express my appreciation to my supervisor, Dominik Bork, for his guidance and feedback throughout the development of this thesis. His input was valuable in helping to refine and shape the final outcome.

I am also thankful to my family and friends, who offered support and thoughtful discussions during the writing process, particularly regarding the structure and organization of the thesis. Their encouragement provided much needed balance during this intensive period of work.

This thesis is grounded in a meta-synthesis approach, which heavily relied on the foundation laid by the work of numerous researchers before me. I am deeply grateful for the collective knowledge and insights provided by the academic community, which made this study possible. Their contributions were essential in understanding the complexities of the topic and framing this research.

Lastly, in the development and laborious completion of this work, I have been grateful for a number of tools, methods and resources that have supported the progress of the work. This work is a product of collective knowledge and collaboration, and I am pleased to have been able to contribute to this topic.

Kurzfassung

Da technische Systeme immer komplexer werden, müssen Unternehmen strategische Ansätze verfolgen, um die gegenseitigen Abhängigkeiten ihrer Prozesse, Werkzeuge und Teams zu verwalten. Model-Based Systems Engineering (MBSE) bietet eine vielversprechende Lösung, doch der Übergang zu MBSE ist ein komplexes Unterfangen, das erhebliche organisatorische Veränderungen erfordert. Diese Arbeit befasst sich mit dem Bedarf an strukturierter Anleitung durch die Entwicklung eines Reifegradmodells, das Organisationen bei diesem Übergang unterstützt.

Auf der Grundlage einer Metasynthese mit einer systematischen Literaturrecherche identifiziert das Reifegradmodell die wichtigsten Herausforderungen, typischen Fallstricke und bewährte Praktiken in vier Phasen der MBSE-Einführung. Dieser strukturierte Ansatz bietet Organisationen Werkzeuge, um ihren aktuellen Reifegrad zu verstehen, Prioritäten zu setzen und häufige Fehltritte zu vermeiden.

Das Modell ermöglicht es Organisationen, die gewonnenen Erkenntnisse auf ihre spezifischen Kontexte zuzuschneiden und gewährleistet so die praktische Anwendbarkeit. Es unterstreicht die Bedeutung von Führung, kultureller Bereitschaft, technischen Werkzeugen, Personalentwicklung und Modellierungspraktiken für eine erfolgreiche MBSE-Implementierung. Darüber hinaus befasst sich die Studie mit bestehenden Lücken in der Literatur, insbesondere in Bezug auf die Priorisierung bewährter Praktiken, und bietet Unternehmen, die einen Wechsel zu MBSE anstreben, umsetzbare Erkenntnisse.

Durch die Reduzierung der Komplexität des überwältigenden und langwierigen Prozesses der Einführung von MBSE und durch die Förderung des Vertrauens in den Übergangsprozess zielt diese Arbeit darauf ab, Organisationen zu einer nachhaltigen MBSE-Integration zu führen.

Abstract

As engineering systems grow increasingly complex, organizations must adopt strategic approaches to manage the interdependencies of their processes, tools, and teams. Model-Based Systems Engineering (MBSE) offers a promising solution, yet transitioning to MBSE is a complex endeavor that requires significant organizational change. This thesis addresses the need for structured guidance by developing a maturity assessment framework that supports organizations in navigating this transition.

Based on a meta-synthesis including a systematic literature review, the maturity assessment identifies key challenges, pitfalls, and best practices across four levels of MBSE adoption. This structured, high-level approach provides organizations with the tools to understand their current maturity stage, prioritize efforts, and avoid common missteps.

This framework empowers organizations to tailor insights to their unique contexts, ensuring practical applicability. It emphasizes the importance of leadership, cultural readiness, technical tools, workforce development, and modeling practices in achieving successful MBSE implementation. Additionally, the research addresses critical gaps in the literature, particularly regarding the prioritization of best practices, providing actionable insights for organizations embarking on MBSE transitions.

By reducing the complexity of the overwhelming and lengthy process of MBSE adoption, fostering confidence in the transition process, this work aims to guide organizations toward sustainable MBSE integration.

Contents

CHAPTER 1

Introduction

Over the past two decades, the landscape of technology has undergone a significant transformation, reshaping the way we live, work and interact with the world around us. This period has witnessed an unprecedented growth of digital innovation, characterized by the rapid evolution of interconnected devices, (big) data analytics and the emergence of sophisticated sensor technologies. As a result, products across diverse industries have transcended their traditional boundaries, morphing into complex ecosystems of interconnected components capable of gathering, processing, and distributing vast amounts of data in real-time.

The increase of data handling devices builds the center of this technological revolution. From smart home appliances and wearable gadgets to industry machinery, autonomous vehicles and aerospace advancements. The progress led to an era of extraordinary connectivity, enabling seamless communication and data exchange between disparate systems.

However, this convergence of technology and connectivity has also introduced an era of complexity, both in terms of product design and manufacturing processes. Unlike their predecessors, modern-day products are no longer standalone entities, but rather one part within an intricate ecosystem of interconnected components, each playing a vital role in the overall functionality, reliability, and performance of the system. The integration of diverse subsystems, coupled with the need for seamless interoperability and data exchange, has significantly increased the complexity of product development and manufacturing processes. The production, depending on the specific device, necessitates a highly coordinated and collaborative approach, involving cross-functional teams of hardware-, software-, electrical-engineering, designers, and other domain experts working together to handle the intricate interdependencies and requirements inherent in interconnected systems.

1.1 Problem statement and research objectives

Due to the aforementioned increased use of technology in many business areas, not only the customers' requirements for multidisciplinary and complex products have increased, but also the complexity of the products themselves. The necessary coupling between disciplines and different business departments, leads to increased cost and difficulty in product development. Model-based System Engineering (MBSE), is defined by the International Council on System Engineering (INCOSE) as

> "... the formalized application of modeling to support system requirements, design, analysis, verification, and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases. ..."

with the purpose to tackle the difficulties of developing complex systems and products [1].

MBSE is often suggested as a solution when facing challenges during the development of complex systems [2][3]. Its effectiveness during product development has been proven in numerous studies in academia as well as in industry, with advantages of improved system understanding, reduction of development time, reduction of errors, increased consistency, and traceability, improved communication, and others [4]. Although the benefits that come with the transition to MBSE are manifold, so are the challenges faced during the adoption phase [5][6].

Breaking up the traditional way of working in a company is not an easy task and the path of this transition is not clearly defined. Several surveys and studies have been conducted with varying outcomes and different best practices that have to be taken into consideration, making it difficult for practitioners and researchers alike to prioritize among them [7]. Currently the research is lacking a holistic perspective on MBSE adoption, bringing together all the mentioned aspects, challenges faced, potential pitfalls and best practices as well as a form of contextual prioritization.

For this reason, the aim of this master thesis is to investigate the adoption of MBSE in companies and address the associated challenges, pitfalls and best practices during the transition from a traditional approach of Systems Engineering to a model-based one. This master thesis seeks to explore the intricacies of this change by conducting a qualitative meta-analysis, also referred to as meta-synthesis, of existing literature. The exploration encompasses a comprehensive review of qualitative studies, case reports, and academic articles to establish a nuanced understanding of MBSE adoption. The study aims to analyze the multifaceted challenges inherent in the shift from traditional systems engineering methodologies, examining the impact of organizational culture, technological infrastructure, and human factors, synthesizing the collective wisdom found in the literature into a synopsis of best practices. These practices, once aggregated and categorized, will offer a guidance for organizations navigating the complexities of adopting

MBSE. Ultimately, this research aspires to provide a holistic overview regarding the adoption of model-based systems engineering by the current state of scientific knowledge, not only seeking to contribute to the ongoing research, but also serving as a practical resource for industry professionals. This meta-synthesis therefore shall respond to the following main research questions:

1. **RQ-1: What are the primary challenges encountered by companies in the adoption of Model-Based Systems Engineering (MBSE), as evidenced in existing literature?**

2. **RQ-2: What are the common pitfalls and best practices in the adoption of Model-Based Systems Engineering (MBSE), and how do these elements vary in priority and application across different organizational contexts?**

3. **RQ-3: To what extent can insights from past MBSE adoption efforts, including identified challenges, pitfalls, and best practices, be synthesized into a maturity assessment framework that guides organizations in the early stages of their MBSE transition by offering a structured adoption plan, and how is this framework received and validated by the MBSE community?**

1.2 Significance and relevance of the study

The significance of investigating MBSE adoption lies in the recognition of existing knowledge and the essential step of synthesizing diverse perspectives, overcoming the confusion about similar recommendations with simultaneously contradicting ones that often arise [7]. While literature on MBSE in general is abundant, varying viewpoints, contexts, goals and outcomes necessitate a cohesive examination to establish a new baseline of understanding. By conducting a qualitative meta-analysis, this research combines fragmented insights, offering a comprehensive view of MBSE adoption.

This study is relevant not only to the scientific community seeking to advance knowledge but also to practical users, including companies and firms undergoing transition. By highlighting primary challenges, potential pitfalls, and best practices gathered from literature, this research equips organizations with valuable insights to navigate their MBSE adoption journey effectively. It serves as a practical resource, empowering stakeholders to make informed decisions tailored to their unique contexts.

The significance and relevance of this research extend beyond theoretical exploration, they lie in its potential to drive tangible improvements in industry practices. By processing collective wisdom into actionable guidance within a maturity assessment, this study facilitates smoother transitions to MBSE, mitigating risks and enhancing outcomes.

1.3 Scope

The scope of this thesis is defined by three overarching research questions that explore the complex landscape of MBSE adoption. Through an in-depth analysis of existing literature, this study seeks to consolidate the primary challenges organizations face, identify common pitfalls, and outline best practices critical to navigating a successful transition to MBSE.

Within this framework, the thesis provides a valuable resource for system engineers, managers, and other stakeholders navigating MBSE adoption. Specifically, it presents a maturity assessment model with tailored guidance at each level, designed to help organizations understand their current standing and identify actionable next steps in their transition to MBSE. This approach enables stakeholders to focus their efforts strategically, fostering a smoother and more effective MBSE transition process.

However, it is important to note that this thesis does not provide a step-by-step guide encompassing the selection of modeling language, methodologies, tools, and necessary training. Such decisions are highly contingent upon the unique circumstances of each organization and require careful consideration beyond the scope of this study. Instead, the thesis serves as a strategic resource, offering insights to inform and guide stakeholders as they navigate the complexities of adopting MBSE.

1.4 Outline

Chapter 1: Introduction to Systems Engineering
We begin with an overview of systems engineering and why this approach is gaining traction in modern engineering. The chapter introduces the traditional, document-based approach, explaining key concepts and definitions essential for understanding the thesis. We then explore the complexity of this conventional development process, discussing life cycle concepts, models, and processes. This sets the stage for the introduction of Model-Based Systems Engineering (MBSE), highlighting its three distinguishing pillars: modeling languages, methodologies, and tools.

Chapter 2: Benefits and ROI of MBSE
This chapter examines the potential advantages of MBSE and the motivations for transitioning, despite the inherent complexities and risks. A key focus is the return on investment, as MBSE requires significant early-stage investment and presents challenges in quantifying its benefits. We also identify domains that stand to gain the most from adopting MBSE.

Chapter 3: Methodological Approach
Here, we detail the research methodology, documenting every step of the meta-synthesis and referencing relevant guidelines. This chapter ensures scientific rigor, allowing for transparency and reproducibility of our approach.

Chapter 4: Results and Discussion
This chapter presents the synthesized findings from the systematic literature review, structured into a four-level maturity assessment. Each level outlines key challenges, pitfalls, and best practices identified from past projects, providing a prioritized list for companies transitioning to MBSE. To enhance usability, the chapter includes simplified images of the maturity assessment, along with links to high-quality, full-size versions for detailed inspection and workflow analysis.

Chapter 5: Survey Feedback and Interpretation
This chapter begins by introducing the survey, detailing its design, target audience, and intended purpose. Furthermore, it provides insights on the experience level of the anonymous respondents of the survey. The feedback gathered is then presented, along with interpretations and, where applicable, responses to key points raised by the participants.

Chapter 6: Conclusion, Limitations and Future Research
The thesis concludes by addressing the initial research questions, summarizing key contributions, and discussing limitations. We also propose refinements to the maturity assessment and suggest directions for future research based on our findings.

CHAPTER 2

Overview of Systems Engineering and Model-Based Systems Engineering

The purpose of this chapter is to provide a comprehensive introduction to Systems Engineering (SE) and Model-Based Systems Engineering (MBSE). It shall provide the foundational knowledge necessary for understanding the subsequent discussions and analyses in this thesis. By explaining the terms, core principles, processes, and standards associated with SE and MBSE, this chapter seeks to clarify and unify the understanding of the various aspects underpinning these engineering disciplines.

This chapter is structured to guide readers through the fundamental concepts and terminologies, ensuring that even those without prior expertise in the field can grasp the essential ideas. Through detailed descriptions and examples, it sets the stage for the exploration of the adoption challenges, pitfalls, and best practices of MBSE, which are the central focus of this thesis.

2.1 Challenges in Today's Engineering World

The product development landscape has changed significantly with the increasing complexity of products and services. Some of the main drivers for the challenges faced in today's engineering world are [8]:

1. **Higher System Complexity:** Modern products are increasingly characterized by multidisciplinary and distributed functions, such as mechatronic or cybertronic systems. Additionally, systems of systems (SoS) encompass components that extend beyond purely technical domains to include socio-technical or ecological systems.

2. **Functional Focus:** There is a growing emphasis on realizing required functionality with the necessary reliability, sensitivity, reproducibility, and availability rather than focusing solely on components. Customers now seek solutions based on specific functions rather than buying individual parts and building the solution themselves. This signals a shift from product orientation to service orientation.
3. **Stringent Safety and Security Requirements:** The development of systems must adhere to increasingly strict safety and security standards, driven by legal obligations and industry-specific requirements such as ISO 26262 and ISO 21448 in the automotive sector, or FDA regulations in the medical field.
4. **Distributed Development Teams:** Development efforts are often spread across nationwide or global interdisciplinary teams, adding layers of complexity to coordination and integration.
5. **Agile Development Processes:** Systems require ongoing development and updates based on feedback even after delivery, necessitating agile development methodologies.
6. **Dynamic Feedback via Digital Twins:** There is a growing desire to capture dynamic feedback from systems during their later life cycle stages using digital twins. This information is helpful for extracting insights, predicting maintenance needs, and as a basis for new business models.

These challenges lead to rethinking and redesigning the development process. An efficient approach to managing system and project complexity is Systems Engineering (SE). SE is a well-established methodology, encompassing various processes along with development activities and methods, which will be introduced in the following sections.

To further improve SE and address these challenges even more effectively, models are used to consistently represent the information defined or obtained during the development process. This model-based approach improves the traceability of information flow and enhances the overall development process [8]. This enhanced version of SE, which integrates models of the product and its functions, is known as Model-Based Systems Engineering (MBSE) and will also be introduced in more detail in the next sections.

2.2 Systems Engineering (SE)

Before we can start to understand MBSE we first have to look at the alternative. Traditional Systems Engineering, often referred to as Document-Based Systems Engineering (DBSE) or simply Systems Engineering (SE), relies heavily on document-centric processes. The key difference between these two approaches, SE and MBSE, are the primary artifacts produced in their life cycle activities [3].

The International Council on Systems Engineering defines [9]:

> *"Systems Engineering is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods."*

SE provides facilitation, guidance, and leadership to integrate relevant disciplines and specialty groups into a cohesive effort. This forms a well structured development process that progresses from concept to development, production, utilization, support, and eventual retirement. SE addresses both the business and technical needs of acquirers, aiming to provide quality solutions that meet user and stakeholder needs, are fit for their intended purpose in real-world operations, and minimize adverse unintended consequences [9].

The primary goal of SE activities is to manage risk, including the risk of not delivering what the acquirer wants and needs, the risk of late delivery, the risk of excess cost, and the risk of negative unintended consequences. The utility of SE activities is measured by the degree of risk reduction they achieve. Conversely, the acceptability of omitting an SE activity is assessed by the level of excess risk incurred as a result [9].

Not every project, development process, or company requires Systems Engineering. Its necessity depends on the project's complexity, scale, and specific requirements. However, SE has gained significant relevance in recent years because it improves the management and execution of complex projects compared to non system approaches [9].

2.2.1 Key definitions and concepts

The principle of Systems Engineering has been developed in the middle of the 20^{th} century [10], although its origins lie far further back [9], to effectively manage the rising complexity of projects specifically in the military and aeronautics domains [10].

In order to understand SE, its principles and concepts, we first have to establish an understanding of the terminology used. For this purpose, standards are defined and revised by a collaboration of multiple committees like the Software and Systems Engineering Standards Committee (IEEE), the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

Among many other standards defined in the area of Systems Engineering, the ISO/IEC/IEEE 15288 standard is the international standard for "Systems and software engineering — System life cycle processes" [11]. It is often referenced, when general explanations and definitions in the scope of SE are needed. Contrary to what the name might appear to convey, this standard does not only define the system life cycle processes, but also offers several essential definitions to better understand the domain talked about.

Additionally, the International Council on Systems Engineering (INCOSE) offers a comprehensive SE Handbook [9], which is periodically updated to include the latest revisions and improvements. Each new edition reflects the evolving practices and standards in the field. While the handbook frequently references the standards defined in ISO/IEC/IEEE 15288, its explanations and formatting are more accessible and less rigid than those found in the formal standards documents. This makes the INCOSE SE Handbook an invaluable resource for understanding Systems Engineering, and it serves as a key reference source for this chapter of the thesis.

Definition of System

The formal definition of a system, defined by ISO/IEC/IEEE 15288 (2023) [11] is

> *"A system is an arrangement of parts or elements that together exhibit behaviour or meaning that the individual constituents do not."*

A system can be considered either a product or the services it provides. Its meaning is often clarified with an associative noun (e.g. medical system, aircraft system) or replaced with a context-dependent synonym (e.g. pacemaker, aircraft), although this may obscure the principles of systems thinking. A complete system encompasses all necessary equipment, facilities, material, computer programs, firmware, technical documentation, services, and personnel for self-sufficient operation in its intended environment [9].

SE practitioners focus on systems engineered for a specific purpose. INCOSE defines an engineered system as one designed or adapted to interact with an operational environment to achieve intended purposes while adhering to applicable constraints. Engineered systems may include people, products, services, information, processes, and natural elements [9].

System of Interest (SoI) and System of Systems (SoS)

When working in Systems Engineering, multiple systems often interact with one another, making clear distinctions among them essential. The definition of a system alone is insufficient, as the focus on a particular system can change depending on the viewpoint or stakeholder, as specified by [11]:

> *"The perception and definition of a particular system, its architecture and its system elements depend on a stakeholder's interests and responsibilities."*

Understanding the distinction between systems and their boundaries is fundamental to comprehending system behavior. A system encompasses both internal and external perspectives, where the system boundary separates the system of interest (SoI) from its environment or context. The environment includes all external elements that interact with the system, such as users and other supporting systems throughout the system's life cycle. The functionality of a system is derived from its interactions within the system and with its environment, shaped by the organization of system elements [9].

Emergent behavior is a fundamental characteristic of systems, referring to properties or behaviors that only make sense when attributed to the system as a whole, rather than its individual components. These emergent properties arise from the interactions and interdependencies among system elements. They can manifest as either desirable outcomes, such as synergy and functionality, or undesirable effects, such as inefficiencies or failures [12]. Effectively managing emergent behavior requires system engineers to analyze not just the individual components but also their interactions and the collective dynamics of the system at a higher level. This holistic perspective is essential for designing systems that achieve desired outcomes while proactively identifying and mitigating potential risks associated with undesirable emergent properties.

Now that the defining characteristics and behaviors of a system-of-interest are defined it is important to note, that a SoI for one stakeholder might be viewed as a system element or part of the environment by another stakeholder. Additionally, a SoI can be a constituent system within a system-of-systems (SoS).

As the saying goes, *a picture is worth a thousand words*, consider this illustrative example from [9] to exemplify the concept of a SoS in Figure 2.1:

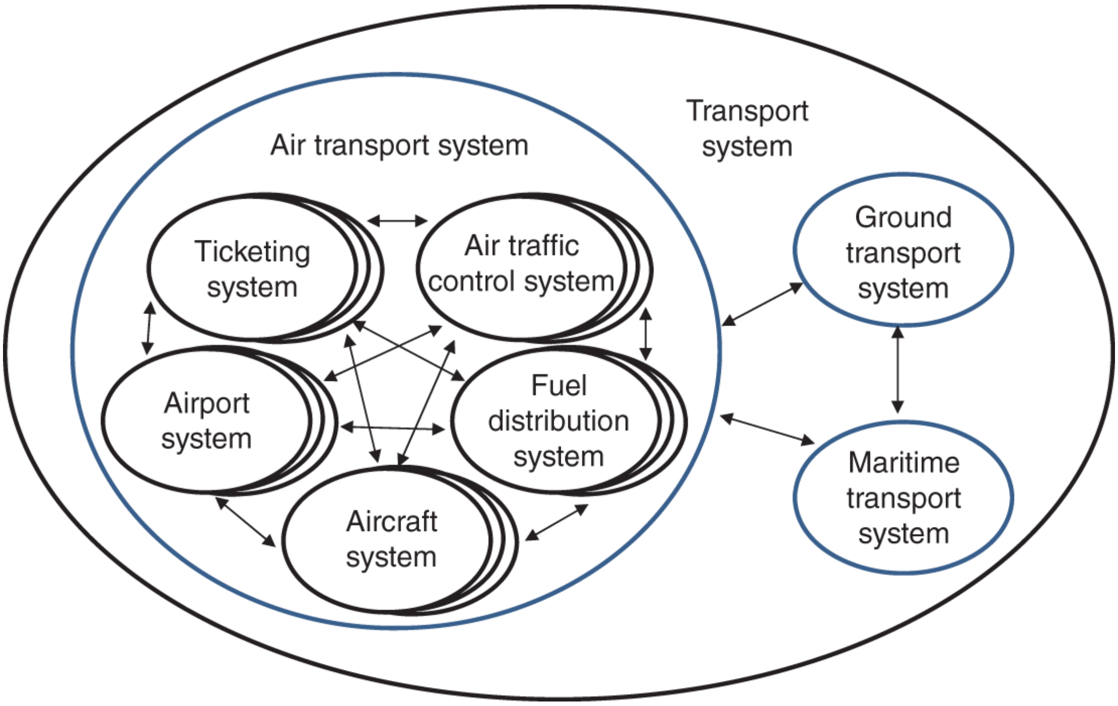


Figure 2.1: Example of systems and systems of systems (SoS) [9]

The air transport system is a SoS comprising multiple aircrafts, airports, air traffic control systems, and ticketing systems, which along with other systems, facilitate passenger transportation. Equivalent ground and maritime transportation SoSs contribute to the broader transport system, forming an interconnected web of systems.

This demonstrates clearly, how constituent systems can be part of one or more SoS. Where each constituent system is a useful system by itself, having its own development, management goals, and resources, but interacts within the SoS to provide the unique capability of the SoS, which none of the systems could accomplish on its own [9].

Systems Engineering, Systems Thinking and Systems Approach

The purpose of SE is to conceive, develop, produce, utilize, support, and retire the right product or service within budget and schedule constraints. Achieving this requires a common understanding of the current system state and a unified vision of future states, transforming stakeholder needs, expectations, and constraints into viable solutions. The right product or service is one that meets the required mission objectives [9].

SE is particularly crucial in managing complexity, which has become increasingly prevalent in modern systems. These systems are often formed by integrating commercially available products e.g. sensors or chips or independently managed and operated systems to deliver emergent capabilities, inherently increasing complexity.

Complexity in systems presents unique challenges. Simple, complicated, and complex systems require different approaches. Simple systems have elements with readily comprehended relationships, while complicated systems have elements with relationships that can be unfolded and understood, providing certainty between cause and effect. Complex systems, however, have interwoven relationships between elements that are not fully comprehended, leading to uncertainty between cause and effect. These systems can exhibit beneficial behaviors like self-organization but can also present novel, nonlinear, and counterintuitive dynamics over time, resulting in suboptimal operation or unintended consequences. Even relatively simple systems can generate complex behaviors, so complexity does not negate simplicity [9].

Traditional SE processes, suitable for complicated systems, take a reductionist approach, decomposing problems into parts, solving them, and reassembling them into a whole solution. This works well for fixed, deterministic, or predictable patterns of behavior, but struggles in complex environments, such as designing autonomous vehicles or socio-technical systems. A fundamentally different approach is required to fully understand how interactions between system parts lead to emergent behaviors.

Systems science emphasizes the need for a holistic understanding of systems beyond their individual parts, independent of their type, physical, natural, engineered or social. This understanding helps in predicting how systems behave and evolve, making it the theoretical basis of how SE works.

Systems thinking applies the properties, concepts, and principles of systems science to real-world problems. It provides a framework for understanding complex situations by recognizing patterns, relationships, and interdependencies within systems. Systems thinking encourages a holistic view, ensuring that we consider the entire system rather than just individual components [9].

The systems approach, based on systems science and systems thinking, is what makes SE effective. Decomposing a problem into its components and analyzing each part, following the principle of divide and conquer, is crucial to refine the understanding of complicated systems, but without considering the broader context and interdependencies, the decomposition can lead to disjointed and inefficient systems [13].

For example, in an airplane's development, considering the wing, an engineer cannot only look at the design, the mechanical and software functionality and interdependency of these systems, but has to keep in mind that his work influences several other systems and characteristics of the larger SoS (the airplane), e.g. the fuel efficiency, passenger comfort, and maintenance requirements. This would in return influence the overarching SoS, being the air transportation system, as the comfort during the flight would impact the satisfaction of the customers of the airplane (stakeholder), which might lead to the decision to use a different transportation system. A bad fuel efficiency could influence airlines buying the plane (stakeholder) to choose a different manufacturer, either due to bad economic repercussions (waste of fuel = waste of money) or due to environmental strain, which could also lead to passengers choosing a different way of transportation.

A good visualization summing up the described difficulties in developing complex systems is shown in Figure 2.2 by [14], where several blind men look at individual parts of an elephant:

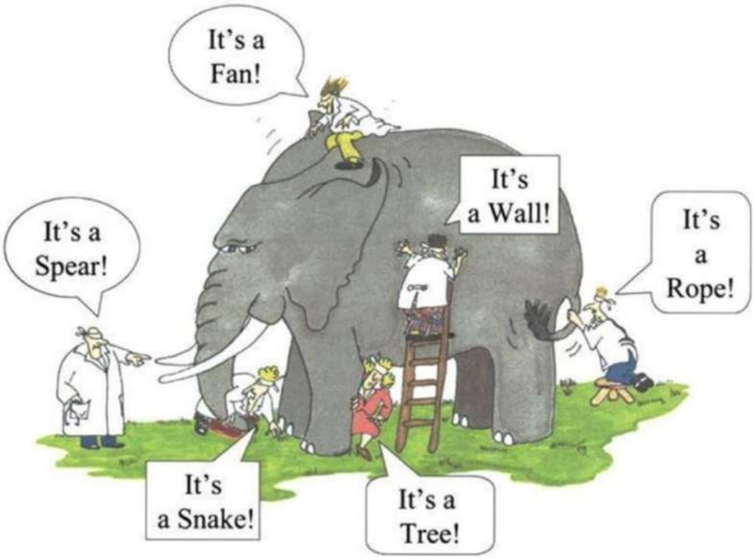


Figure 2.2: Challenge of seeing the bigger picture [14]

Illustrating the challenge of enabling diverse stakeholders to see the bigger picture. Which is precisely the issue, that the holistic perspective offered by Systems Engineering is tackling. Addressing the challenge of modern engineering projects with a multitude of stakeholders and interacting systems. Offering traceability of requirements, ensuring that each one is linked to its origin including rationale, and that changes are managed systematically, enabling better collaboration among stakeholders and improved understanding of system's structure and behavior [13].

With this foundational understanding in place, we can now dive into the life cycle concepts and processes of an SE approach. These steps and life cycle stages are crucial in order to comprehend how SE and MBSE impacts the work and management of the entire lifespan of a product or system.

2.2.2 Life Cycle Concepts

This section will explore the life cycle stages and illustrate how Systems Engineering is practically implemented, from initial concept through to deployment and maintenance. By examining these stages, we want to give a better insight into the systematic methods used to manage complexity.

As already mentioned, the main purpose of SE is to enable the successful realization of a system while balancing competing stakeholder objectives. To achieve this goal, the development effort is broken down into stages, each with specific decision points called decision gates. These gates ensure system characteristics are met, risks are acceptable, and the system is ready to proceed to subsequent stages. It is important to note that life cycle stages can occur sequentially, in parallel, or be revisited as needed, allowing for flexibility and iterative development [9].

Typical Lifecycle Stages

The life cycle stages of a project can be roughly compared to the stages in the life of the average human being. It is not predetermined by any entity which stages a person will go through in life, how long the person will stay in each stage, or if some stages will be repeated. Instead, these stages depend on many factors, such as the family one grows up in, financial circumstances, personal goals, and other life circumstances. However, it is possible to define the typical stages the average human being will experience.

Starting with early development after birth, a person progresses through stages such as infancy and early childhood, where fundamental skills and characteristics are developed. This is followed by the educational stage, where formal schooling begins, and individuals acquire knowledge and skills essential for their future. As they move into adulthood, they enter the career stage, where they apply their skills in professional settings, continuing to grow and adapt. Finally, individuals reach the retirement stage, where they conclude their professional activities and move towards the end of their active life.

Due to the variability of humans, their goals, preferences, and circumstances, these stages can vary significantly in real life. For instance, someone pursuing a PhD will have a much longer educational phase compared to someone entering a different profession straight out of high school. Additionally, one could argue that the educational phase continues alongside professional life, as lifelong learning and continuous education is common in today's world. Similarly, the life cycle stages for the development of different products, within different companies, or organizations such as the Department of Defense or NASA, can vary greatly and also run in parallel.

This is perfectly illustrated by Figure 2.3 in [9]:

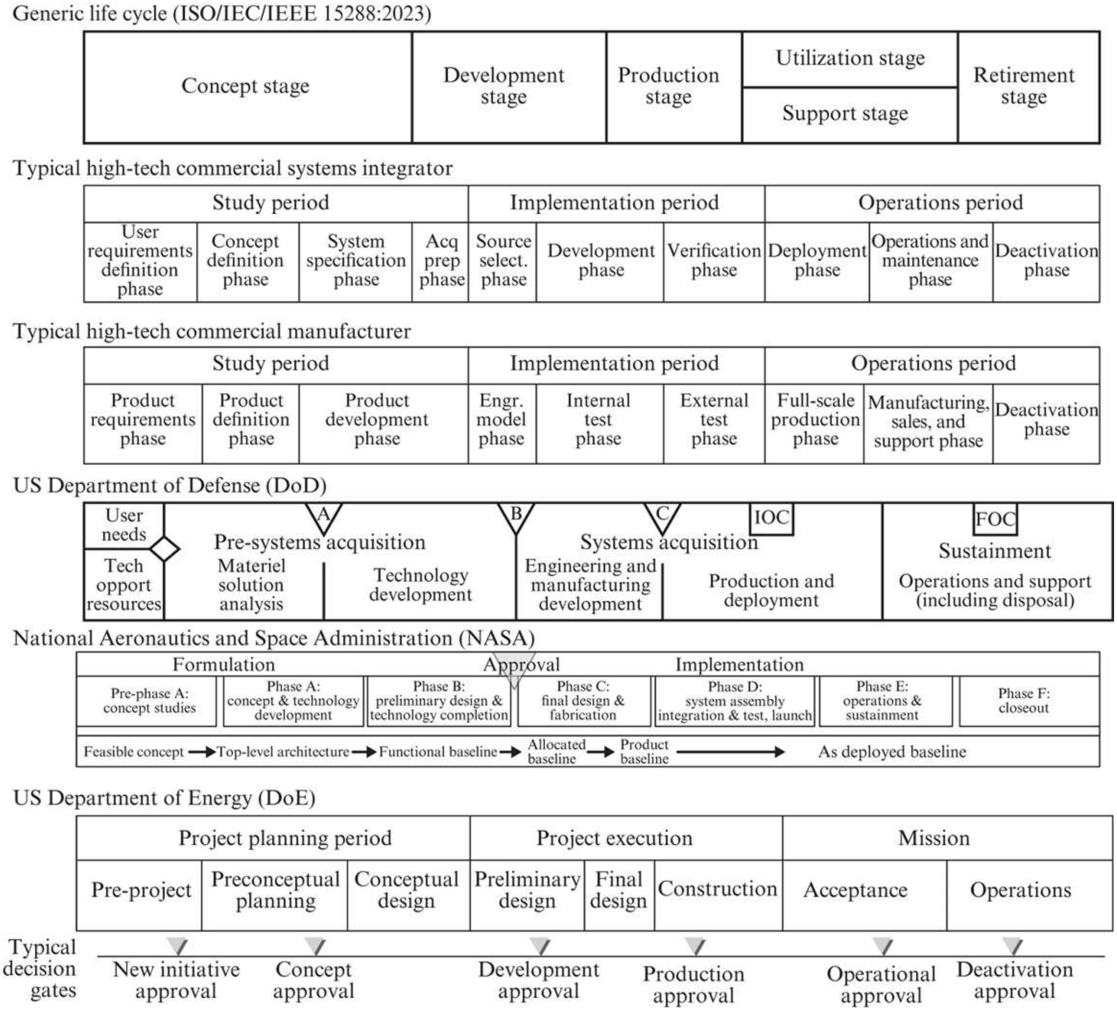


Figure 2.3: Example of varying life cycle stages [9]

Note: This figure is illustrative and not to scale of the relative duration of each life cycle stage. For instance, some systems may progress from concept to deployment within a few years, then remain in service and undergo upgrades for decades. Others may cycle through development and retirement more rapidly [9].

Although the stages, subdivisions, names, duration etc. differ considerably, they all follow the overarching concept of various stages, each with distinct characteristics and objectives to ensure systematic development, efficient integration, and effective realization of complex systems, from initial conception to final retirement.

The generic life cycle stages, defined by ISO/IEC/IEEE 15288, are [9]:

- **Concept Stage:** The concept stage begins with recognizing the need for a new or modified capability. Exploratory research identifies potential solutions through surveys, trade-off studies, and business or mission analyses. The problem space is defined, and stakeholder needs and requirements are identified. Outputs include preliminary concepts, feasibility assessments, and initial requirements. This stage is particularly critical, as the decisions made here will shape, with increasing difficulty to change, the possibilities for all the remaining stages.

- **Development Stage:** In the development stage, the system concepts and stakeholder requirements are refined into an engineering baseline. This includes system requirements, architecture, design, and plans for production, utilization, support, and retirement. Outputs include prototypes, integration plans, risk management strategies, and detailed cost estimates and schedules for future stages. The goal here is not perfection, but to adequately meet the stakeholder needs and requirements in a way that is supportable.

- **Production Stage:** The production stage translates the development baselines into an actual system. This includes the enabling systems and results in the realized SoI and its documentation for use in the subsequent stages. The system is qualified for use and ready for installation and transition to the utilization stage.

- **Utilization Stage:** The utilization stage begins with transitioning the system into its intended environment, where the system provides its intended capabilities. Throughout this stage, which is typically much longer than the other stages, modifications are often introduced in order to remedy deficiencies that appeared, enhance the capabilities or extend the lifetime of the system. During all of this time, it is critical to maintain documentation from prior stages, and ensure processes such as configuration and risk management, as well as SE support are in place and robustly applied. This stage proceeds in parallel with the support stage and possibly ends separately for different parts of the SoI, e.g. if some functionality is replaced or covered by a different system.

- **Support Stage:** The support stage involves provisioning support for the system's utilization. During this time, deficiencies and failures are noted and used as basis for either the remediation of the discovered problem or to start developing an evolutionary modification of the SoI. This type of evolution of the system is often associated with extensive work, but can be necessary to resolve supportability issues, reduce operational costs, or extend the system's life. If the decision is made to realize such an advancement, SE assessments are required to ensure that changes do not compromise system capabilities while under operation or violate requirements. The support stage ends when the system has been judged as at the end of its useful life or the choice has been made that it should be no longer supported.

- **Retirement Stage:** The retirement stage involves removing the system or its elements from operation. SE activities focus on satisfying disposal requirements and archiving documentation from previous stages. Planning for retirement occurs during the concept and development stages, ensuring responsible end-of-life disposal and minimizing long-term consequences.

After exploring the various typical life cycle stages that outline how a system progresses from conception to retirement, it is important to understand the mechanisms that ensure each stage transitions smoothly and effectively.

Decision Gates

This is where decision gates come into play. Decision gates serve as critical checkpoints at the beginning and end of each stage. They help to ensure readiness to continue with the next stage as the current one accomplishes its objectives. The decision gates make sure all the decisions are unambiguously made and documented and relate directly to criteria established to begin or end a particular stage [9].

Typically their goals are to confirm [9]:

1. System maturity is within a defined threshold
2. Project deliverables satisfy the business case
3. Resources are sufficient for this and subsequent stages
4. Issues that need to be addressed in this stage are addressed
5. The overall risk to proceed in the system life cycle is acceptable

As already mentioned, stages do not have to be strictly sequential, but can run concurrently. Which in complex systems can lead to parts being moved forward, while some are on hold and others are terminated or reformed before moving on. The respective decisions are made during reviews with qualified experts, stakeholders and management and should be based on the criteria of the review as the consequences of superficial reviews or skipping a decision gate are often long-term and costly. If the project environment or other factors in the project change significantly, the decision gate criteria should be updated and evaluated.

The successful completion of a decision gate is usually accompanied by several kinds of artifacts (e.g. documents, analysis results, diagrams, etc.) that have been approved as a new basis the future work must build upon. These artifacts are stored in configuration management along with the decisions made and the associated rationale and assumptions for each one [9].

Technical Reviews and Audits

Due to the criticality of the decisions made and the variable length of the various stages in a project lifecycle, decision gates are not the only point where progress is assessed and corrective steps taken or new baselines are agreed upon. Across the different organizations and corporations a multitude of technical reviews and audits can happen throughout specific lifecycle stages. Depending on the corporation, the specific reviews and audits can be tailored to projects or domains [9]. Especially in safety-critical systems, specific technical reviews are essential to baseline artifacts required for certifications necessary for a SoI to be integrated into a SoS. In these contexts, safety considerations are paramount due to the potential impact on human lives e.g. aerospace, automobile manufacturing, spaceflight etc. [15].

While describing all technical reviews and audits across various industries is impractical, given the sheer volume and complexity, these examples should help the understanding of the lifecycle procedures involved. Therefore here are some examples used by NASA's spaceflight missions [16]:

- **Mission Concept Review (MCR):** Confirms the mission need and evaluates the proposed objectives as well as the concept for meeting them.
- **System Requirements Review (SRR):** Evaluates functional and performance requirements defined for the system or project plan and ensures that combined with the concept they can satisfy the mission.
- **Preliminary Design Review (PDR):** Demonstrates that the preliminary design meets all systems requirements with acceptable risks and within cost and schedule constraints. Furthermore, it affirms the correct design options have been chosen, interfaces identified and verification methods described. The PDR should address and resolve critical issues as well as show that work can begin.
- **Critical Design Review (CDR):** Determines if the technical effort is on track to develop a system that meets the performance requirements within cost and schedule constraints.
- **Flight Readiness Review (FRR):** Examines tests, demonstrations, analyses and audits that determine the readiness for a safe and successful flight. It additionally assures that all flight, ground hardware, software, personnel, and procedures are operationally ready.
- **Decommissioning Review (DR):** Confirms the decision to terminate or decommission the system and assures the readiness of the system for the safe decommissioning and disposal of system assets.

These examples were not selected based upon their criticality, but for their clarity and comprehensibility, allowing the general concept of these reviews to be easily grasped.

Based on the knowledge from the previous sections, it should be evident that while these reviews are presented in the typical order of their appearance, they do not all belong to the same life cycle stage.

Additionally, it is important to note that each review has specific timing when it will take place, entry / success criteria, and clearly defined expected outcomes, sometimes even specific artifacts that are baselined as the result of a review [16]. The details regarding the reviews or audits in a specific project or organization are usually captured in the company's or project's Systems Engineering Management Plan (SEMP) and reflected in the defined schedule [9].

2.2.3 Life Cycle Model and Processes

Having established the framework of system engineering work practices, including the lifecycle stages, decision gates, and technical reviews and audits, we now look into the specifics of what actually happens during these lifecycle phases, between the decision gates, audits and technical reviews. This section will provide an overview of the methods and processes employed to achieve the goal of developing a system that satisfies stakeholder needs and requirements.

The overarching concept within which the individual life cycle stages and their transitions are planned and implemented is the life cycle model. The three primary approaches are the sequential, incremental and the evolutionary approach, where each has unique characteristics and is suitable for different scenarios [9].

A detailed understanding of life cycle models is not necessary for grasping development processes and general SE and MBSE practices, but recognizing their differences can improve the comprehension of how complex systems are developed.

Sequential Approach

The sequential approach, often referred to as the "waterfall model" progresses through a series of defined stages in a linear fashion. Each stage must be completed before the next begins, as it depends on the deliverables of the previous one, ensuring a structured and predictable workflow [9].

Especially in the context of safety-critical products a sequential approach has its benefits as in order to meet modern certification standards a thorough, documented set of plans and specifications has to be followed. These standards mandate strict adherence to process and specified documentation to achieve safety or security [9].

This model is well-suited for projects with well-understood requirements and low levels of uncertainty. An aspect that is especially important in this approach is change management. Changes to enhance system performance or to reduce risk or cost are welcome for consideration, but after baselining artifacts, change requests must go through formal change control, since others may be building on previously defined and released design decisions, making the outcomes predictable, but the development approach not susceptible to change [9].

Incremental Approach

The incremental approach breaks down the system development into smaller, manageable increments or modules. A project starts with an initial capability or set of capabilities followed by successive deliveries, where each one builds upon the previous, gradually adding functionality until the complete system is realized. This approach allows for early partial deliveries, planned intervals with new versions or capabilities and can as a consequence accommodate changes more easily than the sequential model [9].

Evolutionary Approach

The evolutionary approach emphasizes iterative development and continuous refinement. In contrast to the sequential and incremental approach, the evolutionary approach does not assume that the full set of functionality is mostly known at the beginning of the project effort. This approach provides the adaptability and flexibility for these kinds of situations or novel systems. Here, systems are developed through repeated cycles (iterations), allowing for frequent reassessment and adjustment. It is particularly important that the experience gained from earlier iterations is transferred to subsequent ones. Especially when multiple versions are operated and supported simultaneously a well-functioning configuration control is important as it will otherwise lead to confusion and negative impact on cost and schedule [9].

The significant advantage this approach offers is the possibility of steady and high-quality feedback from relevant stakeholders. Early versions can be used to demonstrate feasibility and building the minimal viable product (MVP). Adding, updating or removing functionality with following updates. This makes the model particularly useful for projects with high uncertainty or rapidly changing requirements supporting flexibility and adaptability [9].

Selecting an appropriate life cycle model for a specific system or project involves considering various factors, the stability and variety of environments, novelty, risks concerning the stakeholders, etc. It is also possible for certain cases to combine approaches in order to make use of both characteristics. This can be useful in situations where the teams working on the project such as electronic hardware, firmware, or software teams have different timings for their versions [9].

System Life Cycle Processes

The system life cycle processes include the steps that actually happen during a project. A process is defined as a series of activities and tasks performed to achieve one or more outcomes for a stated purpose [9]. The processes encountered in SE are enablers to help manage a system solution across all lifecycle stages. These processes, just like the higher-level life cycle stages, do not simply happen strictly sequentially. A big factor what makes the management of these processes throughout the systems development so difficult, is the fact that they are applied concurrently, iteratively as well as recursively in combination with other enablers like various tools [9].

It is essential that valuable information and insight can be exchanged between processes to be able to ensure a good system definition that effectively and efficiently meets the stakeholder needs and requirements. Concurrency, iteration and recursion account for this ongoing learning and updated decisions. They allow the incorporation of results from analysis and other process applications as the solution evolves. To ensure a common understanding of these terms, here is a simplified explanation from [9]:

- **Concurrency** in system engineering involves the **parallel application of processes at different levels in the system hierarchy**, allowing multiple processes to occur simultaneously when they are not dependent on each other for information or results. For example, the Risk Management and Measurement processes typically run concurrently, providing information to one another continuously.
- **Iteration** involves the **repeated application and interaction of two or more processes at a given system hierarchy level** to accommodate stakeholder decisions, evolving understanding, architectural decisions, and trade-offs for aspects like affordability and feasibility. For example, there is often iteration between the System Requirements Definition and System Architecture Definition processes. The evolving requirements can shape the architecture, such as when electrical modeling shows a system element's load exceeds its power budget, necessitating design changes. This iterative process continues with further analysis and trade-offs to refine requirements and architecture.
- **Recursion** involves the **repeated application of life cycle processes at successive levels of the system hierarchy**. Each level uses tailored processes, where outputs from one level become inputs for the next. This continues until a decision is made to make, buy, or reuse a system element. For example, outputs from system definition at one level become inputs for system realization at the next.

It is apparent by this explanation, without looking at the specifics of all the processes involved, that coordinating processes, process steps, iterations, and interactions of different teams to deliver satisfying results, is highly complex and requires a lot of effort. This complexity is a significant challenge for traditional SE approaches, and it is a frequently cited argument for adopting MBSE [17][18]. We will further explore the reduction of complexity and other benefits attributed to MBSE in section 3.1.

Moving on to the processes happening during the application of SE, there are four process groups defined by [11]. The groups are agreement processes, organizational project-enabling processes, technical management processes and technical processes. All of the included processes together build the backbone of how companies operate in the area of systems engineering.

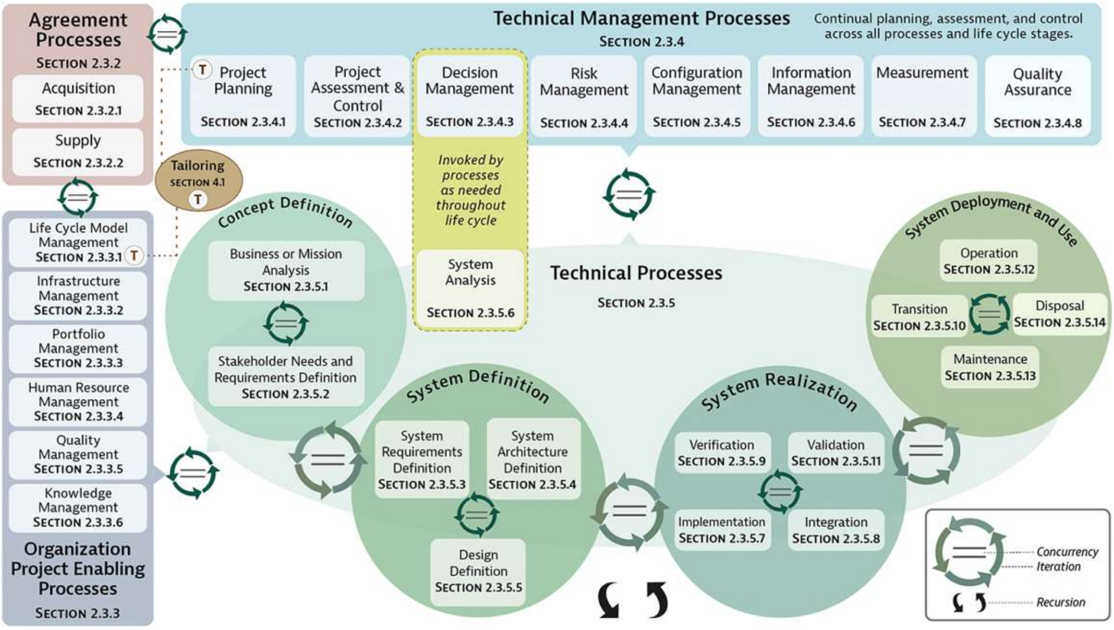


Figure 2.4: System life cycle processes [9]

Looking at Figure 2.4 from [9], we can observe the interaction between the process groups and gain a better understanding of the procedures involving concurrency, iteration and recursion.

This overview is intended to familiarize non-engineering experts with the processes involved in developing complex products in today's world. The degree to which companies or organizations implement these processes depends on various factors, including budget, timeline, personnel, and other constraints. In contemporary engineering practice, it is common to adapt and tailor processes to fit the specific needs of a project or to align with the established standards of a particular company.

Therefore, the information presented here should not be interpreted as an obligatory or universally applicable guideline, but rather as a representative explanation of the processes typically involved when using a Systems Engineering (SE) or Model-Based Systems Engineering (MBSE) approach. For additional insights into the processes, we encourage readers to refer to the respective sections (visible in Figure 2.4) in [9], where these processes are described in exceptional detail.

The focus now shifts to the technical processes, as these are the areas where the transition from SE to MBSE becomes most evident and impactful. Their inclusion aims to enhance understanding of how these approaches function and contribute to the development of complex systems. This will become more apparent, when we take a closer look at the value an MBSE approach brings to theses processes in section 3.1.

Technical Processes

All together there are 14 technical processes that can all be seen in Figure 2.4. As stated by the ISO/IEC/IEEE 15288 standard [11] these processes are used to:

> *"... define the requirements for a system, to transform the requirements into an effective product, to permit consistent reproduction of the product where necessary, to use the product, to provide the required services, to sustain the provision of those services and to dispose of the product when it is retired from service."*

The following processes allow SE practitioners to coordinate the interactions between engineering specialists, other involved disciplines, acquirers, operators, manufacturing / production and other system stakeholders. Without these technical processes the risk of project failure would be significantly higher. Thanks to Figure 2.4 we gain an understanding of how all of the process groups and involved processes work together on a project [9].

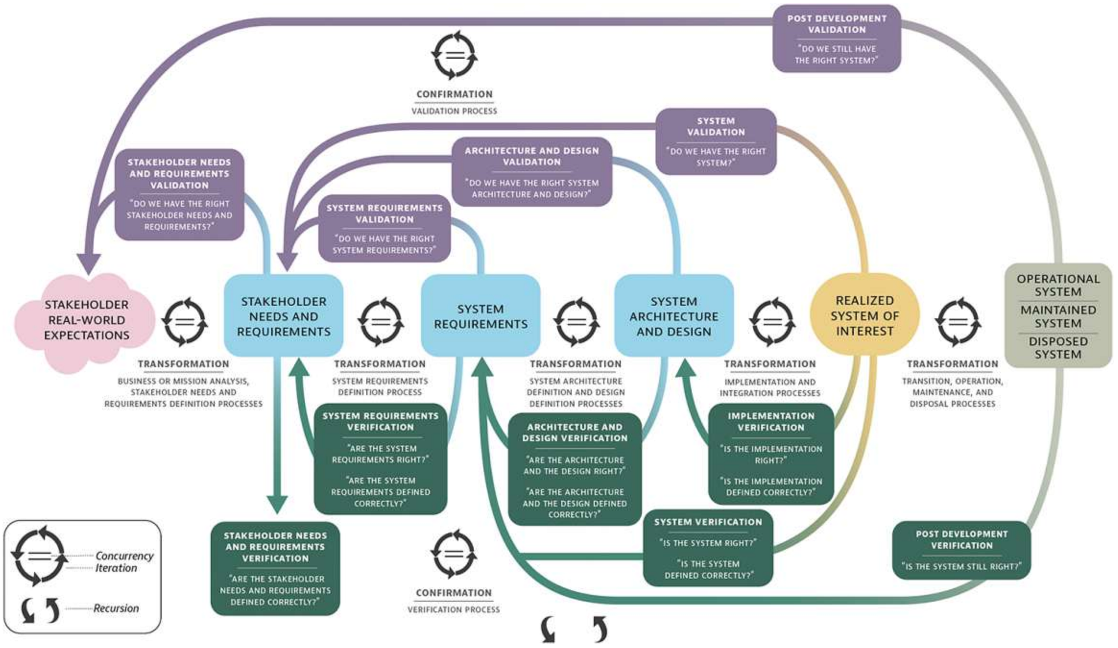


Figure 2.5: Technical Processes [9]

Figure 2.5 provides a good overview of the steps involved in the development of system elements, emphasizing the critical validation and verification between every step. It highlights the extensive work and effort required for system development before it becomes operational. Notably, the figure applies to each system element within the system architecture, meaning that every element goes through these steps until they are integrated into the realized SoI. Further underlining the effort and rigor necessary for successful development [9].

Once the system is operational and maintained, it is essential to regularly assess it through verification and validation to ensure it continues to meet real-world stakeholder expectations and confirm that requirements or capabilities did not change. Should the overarching SoS change, it may necessitate modifications or additions to the SoI's requirements [9].

To finalize the introduction to the SE and MBSE workflow, we will look into the technical processes from [11] [9]:

1. **Business or Mission Analysis Process**

   The Business and Mission Analysis initiates the life cycle of the SoI. It involves identifying and addressing potential problems and opportunities resulting from changes in an organization's strategy and Concept of Operations. This includes developing new solutions or identifying and addressing gaps or deficiencies in existing capabilities. The process consists of defining the problem space, characterizing the solution space, and identifying appropriate solution alternatives. It involves planning for necessary enabling systems or services, ensuring access to them, and analyzing relevant factors such as costs, risks, and performance improvements.

   Through this process, preliminary life cycle concepts are defined, including the acquisition, development, deployment, operations, support, and retirement of the solution. This initial concept is worked on and refined throughout the life cycle as a result of feedback obtained from performing the other technical processes. This feedback and validation are crucial for maintaining alignment with real-world stakeholder expectations. Overall, as the term "business" in the name already implies, the focus is set on identifying and prioritizing business needs, requirements, and success measures that are used to ensure the chosen concepts align with organizational goals.

2. **Stakeholder Needs and Requirements Definition Process**

   After identifying a problem or opportunity and beginning to work on the preliminary concept it is crucial to identify the stakeholders. This process in system development involves recognizing individuals or groups who are impacted by, can influence, or support the life cycle of the SoI. Stakeholders typically include customers, users, operators, maintainers, decision makers, regulatory bodies, developers, validators, and support organizations, including those from external and enabling systems. Due to the amount and diversity of stakeholders the identified needs and later the resulting requirements can be quite extensive.

   Both, needs and requirements, are fundamental concepts in systems engineering, and understanding their distinction is crucial for successful project outcomes. Needs are high-level, broad objectives or desires that stakeholders have regarding a system or product. They represent the underlying motivations or problems that the system is intended to address and are often qualitative, abstract, and not immediately actionable [19]. For instance, in the context of an airplane, a need might be to enhance passenger comfort during long flights. This need is broad and describes the *what* rather than the *how*.

Needs are often related to the customer, technical standards, constraints of the technical domain or manufacturer's practice or innovation targets. While customer expectations and innovation targets often have the highest priority due to their strategic effects, they usually are negotiable. Technical standards on the other hand must be fulfilled without exception, due to product liability reasons [19].

Requirements are then written by carefully considering the needs previously identified. They follow a precise syntax to ensure that they are clear, simple, and never ambiguous. To achieve this clarity, requirements often adhere to the SMART criteria: they must be Specific, Measurable, Achievable, Relevant, and Time-bound [19].

This way the engineers working on the project have clear instructions they have to accomplish in order to make sure the product is being built in accordance to the expectations. The requirements drive and constrain the solution space and thus the majority of the system life cycle technical processes, identifying and characterizing the operational environment and the interfaces needed to external systems for interaction. Since the requirements have such a big impact on the resulting solution, this process is conducted at the beginning of the development cycle, but continuously revisited while the project moves through the other activities.

For this reason it is important to establish traceability throughout the process. The goal is to develop preliminary concepts based on the identified stakeholders, their needs and the resulting requirements. Traceability capturing the rational behind past decisions, from identification of needs to resulting concepts, stored in a project database, is essential for supporting future decision-making and updates.

3. **System Requirements Definition Process**

At this step the identified stakeholder requirements are translated from their user-oriented view of desired capabilities to the technical view of a solution that is able to meet those operational needs. In other words, the stakeholder requirements define what they require from the SoI and the system requirements represent what has to be considered during architecture definition and design definition. The system requirements are typically more detailed and it is common that multiple system requirements result from one stakeholder requirement [9].

Due to the sheer amount of requirements resulting from this and the previous process it is common practice to classify into specific categories. A well-known classification are the following three [19]:

- **Functional requirements:** answer what the system shall do in operation, identifying functions, flows of activities and behaviors.
- **Operational requirements:** answer what kind of usage is foreseen for the system, identifying the external actions that will be applied and the main use cases of the system.
- **Constructional or Architectural requirements:** answer what the system shall be made of, focusing on the decomposition and the configuration of the system.

Figure 2.6 offers a good summary of this classification [19]:

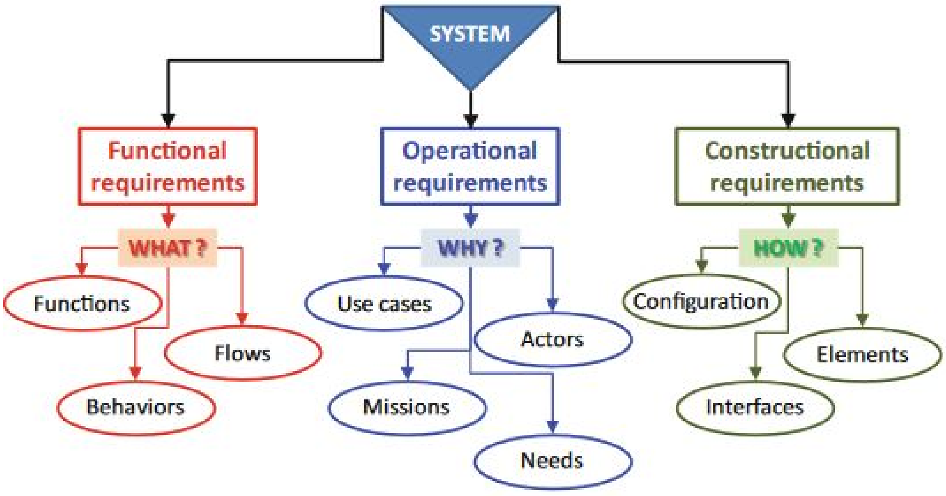


Figure 2.6: Main classification of requirements [19]

4. **System Architecture Definition Process**

   This process transforms the various gathered inputs into one or more system architectures, that address stakeholder concerns and systems requirements and expresses them in a consistent manner. It focuses on the essential concepts, the structure, behaviors and features of the system solution. It additionally supports the understanding of the relations between requirements for the system and the emergent properties and behaviors of it that arise from interactions and relations between the system elements.

5. **Design Definition Process**

   During the design definition process the previously defined architecture is transformed into an actual design of a system, with sufficient detail to allow its realization. An effective architecture allows for maximum flexibility during the design process. The design definition process additionally provides feedback to the architecture definition process to change or confirm the allocation, partitioning, and alignment of architectural entities to system elements that make up the system.

6. **System Analysis Process**

   The System Analysis process integrates with various SE processes to address different dimensions of system evaluation. It supports Business or Mission Analysis by estimating the effectiveness, feasibility, costs, and risks of candidate operations and business models. It aids Stakeholder Needs and Requirements Definition as well as System Requirements Definition by resolving conflicts and assessing feasibility, performance, and technical risks. In System Architecture and Design Definition, it evaluates architectural and design options for feasibility, effectiveness, and critical qualities like dependability and affordability. For Verification and Validation, it quantifies cost, schedule, and uncertainty across different methods. In Project Planning and Assessment, it provides estimates of system metrics against targets and thresholds. Results from these analyses inform decision makers by presenting data for selecting optimal alternatives based on value, risk, and time preferences. Additionally, the process includes cost analysis, assessing life cycle costs, and technical risk analysis, evaluating operational system risks. Effectiveness analysis evaluates how well the system meets criteria in its operational environment, aiding in alternative evaluations and trade-offs.

7. **Implementation Process**

   This process transforms system elements into realized products making them ready for integration. It typically involves activities like fabrication, coding and assembly of the system components based on design specifications.

8. **Integration Process**

   Here, the individual system elements are combined into a complete and functioning system. The integration process also involves ensuring that the components interact correctly with each other and form a cohesive, functioning whole.

9. **Verification Process**

   The verification process checks that the realized system meets all the specified requirements. This is done via inspections, analysis, demonstrations and testing to confirm that the resulting system performs as expected. In simple terms verification is often said to ensure that the "artifact or entity has been built right".

10. **Transition Process**

    This step moves the system from the development environment to the operational environment. Preparing the system for deployment and making sure it can be effectively used by the intended users.

11. **Validation Process**

    This process, as the names says, validates that the system can indeed fulfill the stakeholders needs and expectations. The validation is performed in the real-world operational environment. In simple terms validation is often said to ensure "the right artifact or entity will be or was built".

12. **Operation Process**

    In this process the system is used in its intended environment and delivers its intended functionality. The focus here is on monitoring and management of the system to ensure the correct and efficient operation.

13. **Maintenance Process**

    The maintenance process makes sure the system stays functional and performs optimally over a long time period. This includes regular updates, repairs or improvements to address issues or improve functionality.

14. **Disposal Process**

    Finally, the disposal process is focused on safely decommissioning the system at the end of its life cycle, making sure the no other systems rely on the one being disposed, removing or recycling data and components.

Evidently, the processes from implementation to disposal have been simplified here because they are not as critical for the general understanding required in this thesis. However, they are still included for the sake of completeness and briefly described to provide a full picture of the systems engineering life cycle.

Having established a comprehensive understanding of SE, including its key definitions, concepts, life cycle models, and processes, we now shift our focus to MBSE. It is essential to recognize that the structured approach we just introduced, with its detailed processes, represents the core contribution of SE. This structured framework formalizes holistic systems thinking into a documented and systematic practice that organizations can adopt to improve their workflows. MBSE builds upon this foundation, retaining the relevance of the processes and concepts just covered, while introducing enhancements to how these are executed and their outcomes. By leveraging models as central artifacts, MBSE addresses some of the limitations of traditional SE, refining and extending its capabilities. The following section will explore how MBSE adapts and evolves the SE framework to meet the demands of modern complex systems.

2.3 Model-based Systems Engineering (MBSE)

Now that we have looked at the alternative to MBSE extensively, we have to understand the MBSE approach. The key difference between the two approaches are the primary artifacts created during those activities [3].

During the document-based SE approach the systems engineers will manually generate a large amount of documents throughout the whole lifecycle. Some of those artifacts are for example, the concept of operation, requirements specifications, requirements traceability reports, verification matrices, interface definition documents and many more. Using this approach all of those artifact are in the form of disjoint sets of text documents, spreadsheets, diagrams and presentations [3]. These documents are worked on iteratively, concurrently and recursively while being changed, updated and revisited during work activities, meetings, or specific technical reviews.

Consider this adapted example from [3]. Imagine a system architect working on the fourth iteration of their design, deciding to split up some part of the design, as it will be able to better display the SoI. Now, at least one of the two specific elements has to be renamed. In order to implement this change completely and consistently, the engineer has to locate every text document, table, matrix, diagram or presentation that mentions this system element, access each one sequentially after locating them at the different file servers, intranet websites or configuration management repositories and manually type the change in the respective artifact.

This approach is time consuming and error prone. Many things can go wrong even in this simple example. The engineer may not be able to locate all of the artifacts where the change has to be refactored. The person will therefore likely miss some of them or it could happen that a typo, from manually retyping the change, happens and either way, the affected artifacts will become inconsistent with the rest. This can create a magnitude of problems as other development teams rely on these documents as input for their stage or process of the life cycle, so errors made in the specification will be introduced in the designs of the engineers of the other domains [18]. Furthermore, from a management perspective, this can be problematic, as they have to adjust for increased life cycle cost to fix the resulting defects, which will become higher the further they get propagated down the line [3].

System failures in modern, complex systems are often the result of similar scenarios, with insufficient communication between stakeholders and outdated, incomplete or inconsistent specifications and requirements. These difficulties are exacerbated by the document-based approach due to its point-to-point communication with no mechanism to enforce consistency or completeness between artifacts [18].

Using the MBSE approach, the same life cycle activities are performed also the same set of deliverables are produced. These deliverables are not the primary artifacts of the activities though. With MBSE the primary output is an integrated, coherent and consistent system model [3]. This model serves as a single or authoritative source-of-truth storing all of the model elements and their relationships [20][18][21]. Using this central system model, discipline specific views, meaning they only include information that is specifically relevant for one group of engineers (e.g. system behaviour, software, hardware, safety, security, etc.) can be created [20]. These views can have the form of diagrams or autogenerated textual outputs, but the key takeaway is that they are merely views. They represent some specific aspect of the underlying system model. If a model element is changed in one view or within the model repository, that change is captured and reflected in all other views of that system element [18]. This creates the possibility to programmatically validate the system elements and remove inconsistencies [20].

Considering the example scenario from above, the engineer who had to refactor a system element in the fourth iteration of the design process. Updating this change in an MBSE approach is way simpler. The engineer has to locate the element in the system model and change its name by typing it once. Upon saving, the modeling tool will automatically propagate this change to all diagrams that include this element. If some engineer from any group automatically generates a text artifact, this change will already be included. There is no room for inconsistency between the various views of the model [3].

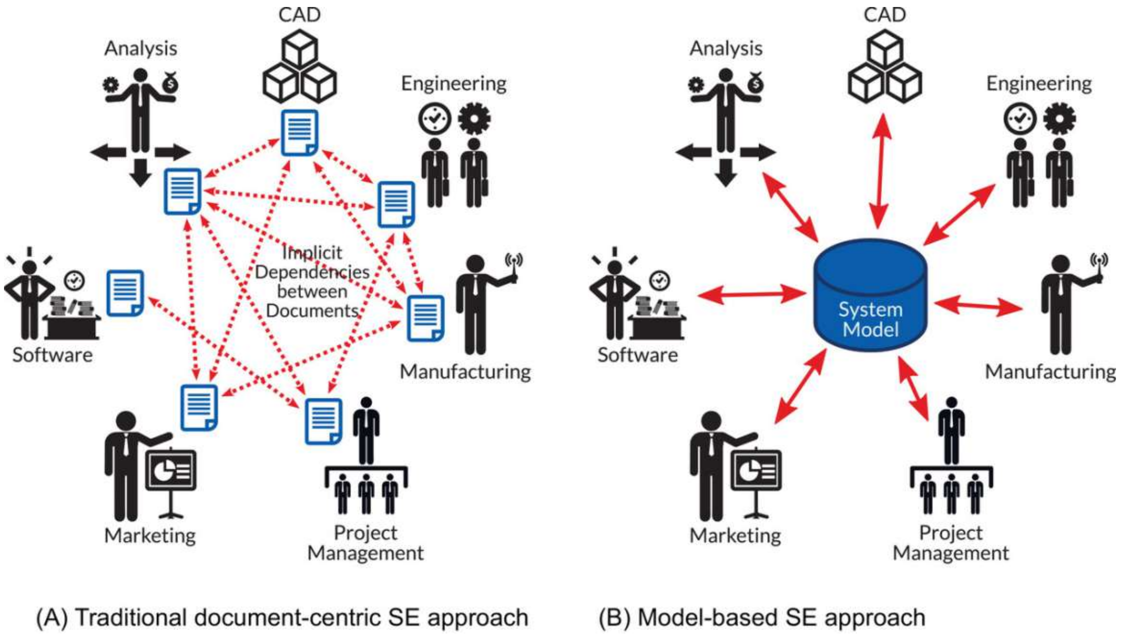


Figure 2.7: Document-based SE approach vs Model-based SE approach

Figure 2.7 from [18] illustrates the above described reality very clearly. In document-based SE there are hundreds of documents sent around, while in MBSE every person that requires information about the system being built knows it will find the most up to date version of it in the system model and can instantly work with it without the fear of inconsistency or requesting and waiting for an answer.

In order to understand what MBSE includes, we have to cover the three pillars that form the foundation of this approach: the modeling language, methodology and tools.

2.3.1 Modeling Languages

Creating a model is like speaking a language, but it is quite different from the natural languages we are speaking in our daily lives [3]. Languages can be roughly grouped into three categories, non-formal languages like natural languages e.g. english, semi-formal languages like diagrammatic languages e.g. UML or SysML and languages based on formal methods enabling formal proofs e.g. Logics or Process Algebra [22][23].

A modeling language is a semi-formal language that defines the kinds of elements and relationships that are allowed in the model. Graphical modeling languages, like the ones used for MBSE, additionally define the set of notations that can be used to display the elements and relationships on diagrams [3].

The most widely used modeling language with regards to MBSE is SysML, which is based on or an extension of UML [24][25]. UML has its origins in software engineering. Due to the rising demand of software in past decades and to improve programming efficiency, object-oriented programming languages, like Java or C++, have been developed. These programming languages encapsulate individuals into objects, which allows to describe their properties and reuse them through inheritance and derivation. UML was then developed to display software code as UML models in order to support software engineering teams when talking about the code [26].

UML, with its roots in software development, defines a notation without prescribing specific methods or approaches for its use. This lack of tool specification and associated methods grants UML a wide application domain across various fields [22]. Similarly, SysML, as an extension of UML, serves as a general-purpose graphical modeling language designed for specifying, analyzing, designing, and verifying complex systems that encompass hardware, software, information, personnel, procedures, and facilities [24]. SysML enhances UML by introducing requirement diagrams that facilitate requirement expression, categorization, and traceability with respect to other analysis and design diagrams [22]. While this flexibility and extensibility make SysML highly versatile, it can also result in semantic ambiguity [24], making it difficult for beginners to start the modeling process without extensive education.

SysML v2 will be the next major revision of the language and improve on the difficulties of its predecessor, with more formal semantics, the addition of textual notation and a standardized Application Programming Interface (API) allowing to quickly access the model data, addressing the problem of model data exchange between tools [24]. This updated version is in preparation, but not yet standardized [22] and will therefore still take a long time before being broadly applied in practice.

The key idea is that each modeling language serves as a standardized medium for communication, with rules that give the model's elements and relationships unambiguous meaning. The ability to construct and read well-formed models is fundamental to the MBSE approach [3].

2.3.2 Modeling Methods / Methodologies

Learning a modeling language is the first important step towards the use of MBSE. The language defines the grammar, the set of rules that determine whether a given model is valid or invalid. These rules however, do not suggest any particular modeling method [3].

When learning about the concept of methodologies in the area of MBSE, there are three important terms [27]:

- **Process:** is a logical sequence of tasks performed to achieve some particular objective. It defines the *WHAT*, without further specifying *HOW* the tasks have to be accomplished.

- **Method:** specifies techniques for performing a task. It defines the *HOW* of each task.
- **Tool:** is an instrument that can enhance the efficiency of a task, when applied correctly with proper training. A tool is supposed to enhance the *WHAT* and the *HOW*.

A methodology, in the context of MBSE, is essentially a recipe defining a set of design tasks including processes, methods and tools, which are used to create a system model or support the discipline of system engineering in a model-based way [3][27].

Similar to how several different modeling languages exist, there are various modeling methodologies around and being developed [28]. They are supposed to be a guide on the MBSE journey and provide a plan that asks the right questions at the right time to drive the process forward. Questions like "What are the expected results of your modeling effort?" or "Do you need to autogenerate text artifacts from the model for review and approval?" or "Which parts of the system need to be modeled?" will help to define the purpose and scope of the modeling efforts that is essential for the MBSE approach [3].

Going into detail on the various methodologies, their procedural differences, application domains, required views, or the criteria for selecting the best methodology for a specific goal, is beyond the scope of this thesis and is still a topic of extensive debate. Numerous comprehensive sources introduce and compare aspects of some widely adopted methodologies like SYSMOD, OOSEM, IBM Harmony and many more [24][27][28][29][30]. Additionally, some sources focus on adapting methodologies to specific needs [29][31] or developing frameworks [30][32] to aid in the optimal selection for individual purposes.

Something often mentioned when discussing MBSE methodologies are System Architecture Frameworks, such as DoDAF (Department of Defense AF), MoDAF (UK Ministry of Defense AF), TOGAF (The Open Group AF), the MagicGrid Framework, and others [33][34]. These frameworks standardize system representations by specifying which views need to be included in the system model [35]. In this sense, they are quite similar to MBSE methodologies, as both address what views should be included. However, they do so with different focuses. MBSE methodologies are primarily concerned with the modeling process specific to systems engineering and the life cycle of the system, ensuring that the models align with engineering activities and requirements. In contrast, architecture frameworks address broader enterprise or organizational architecture concerns, such as ensuring compliance with organizational standards (e.g. Department of Defense requirements). By doing so, architecture frameworks facilitate the integration of system models into larger organizational contexts, supporting strategic decision-making and resource planning [35] [34].

The challenge with MBSE methodologies is that, regardless of which one is chosen, they must be tailored to the specific project to meet its unique needs. These methodologies are typically too broad to be applied directly, as not every step will be relevant to the project at hand, but they do provide a good plan and overall structure [3].

2.3.3 Modeling Tools

The third pillar of the model-based systems engineering approach is the modeling tool. The chosen tool is what enables the developers to construct well-formed models. Usually tools are designed and implemented in a way to comply with the rules of one or more modeling languages. The modeling language specification, such as SysML, is vendor neutral and a specific modeling tool is one vendor's way of implementing that language specification. There exist several commercial as well as free modeling tools for the various modeling languages, that usually vary in cost, capability and compliance with the respective modeling language specification [3].

Modeling tools have to cover many aspects of the product / system life cycle including tasks such as modeling, simulation, system architecting, design, verification, validation, production, delivery, maintenance or project management [36][37], as well as automatic or semi-automatic generation of code from models and test generation [22]. Currently, there is no tool that can support every step of the MBSE approach, but they can be used for certain parts and provide interfaces to other relevant tools that support the remainder of the life cycle [36].

Going into detail on the intricacies and differences of all the available tools is beyond the scope of this thesis. The selection of the best fitting modeling tool, for a specific purpose, is a critical task in the journey towards a model-based approach [21], but there are many aspects that have to be considered and other sources have covered this extensively as well as provided guidance on the process of selection [36][38][22]. An important note in that regard is that the landscape of modeling tools and languages are evolving quickly, so any tool assessment, regardless of its rigidity or discipline, will become outdated quickly [21].

CHAPTER 3

The Case for MBSE Adoption: Benefits and Return on Investment

A simple, yet to the point description of MBSE is given by one of the interviewees of [6]:

> *"... MBSE is the effective use of models to support systems engineering...it's the effective use of models to support your SE process."*

It highlights the fact that MBSE should be considered an extension of SE. The differences between an SE and MBSE approach are not that significant from a conceptual perspective, as all the described concepts from the beginning of this thesis apply to both approaches. They are, however, quite different in terms of how the actual work and the results look like. While SE has proven effective in improving project performance through more repeatable processes, especially in requirements management, architecture, testing, and verification and validation, it has its limits. [39] reports that increased SE effort can reduce cost and schedule overruns, but only up to a certain threshold, beyond which additional SE effort yields no further improvement of the situation. With the complexity of today's engineering challenges, MBSE is emerging as a logical evolution of SE, offering enhanced capabilities to improve on-time delivery and cost management. MBSE provides a range of new opportunities for increased project performance.

Starting the journey towards MBSE is not an easy decision for companies, as it involves overcoming numerous challenges. However, for many organizations, the potential benefits of MBSE outweigh the difficulties and changes required to implement it successfully. The initiation of the transition to MBSE typically falls into two categories: Push and Pull, as described by [40] or Top-Down and Bottom-Up, as outlined by [6].

Push is driven by issues such as past project failures, communication inefficiencies, or market demands for model-based work. This reactive approach arises from the necessity to address and counteract undesirable outcomes. Conversely, Pull is motivated by the potential benefits of MBSE, aiming to improve return on investment (ROI) and enhance the current situation. This approach is proactive, seeking to capitalize on the envisioned advantages of MBSE [40].

The Top-Down approach involves upper-level leadership advocating for and supporting the adoption of MBSE, while the Bottom-Up approach is initiated by engineers or technical personnel who recognize the benefits and strive to promote the idea to upper management [6]. Generally, the Top-Down approach aligns closely with the Pull strategy, as decision-makers at higher levels are often focused on ROI and value creation, actively seeking ways to realize these benefits. In contrast, engineers dealing with inefficient communication and other operational challenges are more likely to adopt the Push strategy, addressing apparent issues and pitching solutions to leadership.

Regardless of how the transition to MBSE is initiated, there must be sufficient drivers, resolvable problems or realizable benefits to convince management to commit to the change. The next section will explore the most frequently cited reasons in the literature for why organizations decide to adopt MBSE.

3.1 Reported Values and Benefits of MBSE

When researching MBSE, the numerous benefits and added value of this approach are frequently highlighted. The following list provides an overview of the many benefits reported by practitioners. While this list is not exhaustive, it offers a clear impression of the improvements many hope to achieve. The effectiveness of MBSE in delivering these benefits can vary based on several factors. For a more detailed list of benefits, refer to [4], which surveyed 240 people and analyzed 360 literature sources that mentioned at least one benefit, serving as the foundation for this collection. Several sources are additionally mentioned throughout the list which we encountered conducting our own research.

[4] categorizes the encountered benefits into four distinct groups: quality, velocity / agility, user experience and knowledge transfer, based on the primary area of impact. These categories are not rigid, as improvements in one area often lead to benefits in another. For example, enhanced traceability not only contributes to better quality but also improves the user experience for engineers who need to trace the impact of changing a requirement.

1. **Quality:**
    - **Reduce errors / defects:** One of the primary benefits of MBSE is the reduction of errors and defects or more precisely the reduction of the impact of errors and defects. The overall number of errors encountered may increase, but they are identified early in the lifecycle, so MBSE prevents these errors

from escalating into more costly problems later on. This proactive approach not only improves overall system quality but also leads to significant cost savings, as the expense of correcting defects increases exponentially the later they are discovered. Thus, MBSE enhances program success by mitigating risks and eliminating undesirable consequences before they can impact the project [21][41][18][17][42].

- **Improved traceability:** Traceability provides a clear and comprehensive linkage between requirements, their justifications and the derived constraints, facilitating a better understanding of the context and rationale behind each requirement. In MBSE, traceability is achieved by examining the hierarchical relationships between parent and child concepts [43][18][44][17][42][45]. A good example is given by [43] that demonstrates the improvement on traceability through MBSE quite well. They show how MBSE allows to precisely track requirements to system elements that satisfy them. This allows to make a conscious decision on whether to remove unnecessary system elements from the SoI. The removal can free up resources like weight, size or power, avoid additional security risk or maintenance, or make the conscious decision to keep it, if that change would significantly impact the system and lead to a lot of rework. Through this traceability the decision can be made based on the cost and risk associated with removing or retaining the component.
- **Completeness:** The use of MBSE greatly enhances completeness by generally improving analysis capabilities [41][42][46], which leads to a more comprehensive set of requirements. It mitigates requirement volatility that can significantly increase engineering efforts and costs in later phases. MBSE ensures better traceability and correct specifications, unlike manual processes in DBSE, and allows for a deeper understanding of functional behavior. It also facilitates the derivation and refinement of constraints and requirements, tracking their maturity and completeness as the model evolves. This results in more accurate and complete system specifications from the early stages of the lifecycle [39][21][44].

2. **Velocity / Agility:**

- **Consistency:** One of the most well-known benefits of MBSE, already discussed in this thesis, is its enhancement of consistency by serving as a Single Source of Truth (SSoT). All system engineering data is stored in this centralized model repository, which contains model elements and their relationships, ensuring that any changes made to an element are automatically reflected across all views and outputs. Unlike the document-centric approach, where inconsistencies often arise due to parallel updates and fragmented information, MBSE maintains consistency by updating all references of an element simultaneously. This method not only prevents duplication of information but also ensures that all SE artifacts remain consistent despite inevitable changes in requirements, constraints and designs, thereby providing a cohesive and accurate representation of the evolving system design [39][21][18][44][45][47].

- **Verification and Validation (V&V):** MBSE's benefit in terms of verification and validation is its ability to specify verification methods early in the program, enhancing thorough planning and providing detailed traceability between requirements and test plans. Unlike traditional engineering processes, where automation of V&V is challenging at early stages, MBSE enables early and continuous V&V through model-based simulations. This allows the system's performance to be analyzed and evaluated before production, providing valuable evaluation information for later phases [39][36][42][21].
- **Concept exploration:** Using an MBSE approach for developing mission architectures allows organizations to explore a broader set of design options within the same time and resource constraints as conventional methods. The resulting more informed decisions can significantly improve mission reliability (reduce risk) and prevented later costs due to potential rework or system failure [39][41][42].
- **Design reuse:** MBSE facilitates significant design reuse across product lines by enabling the storage and management of design elements as reusable model components. This capability allows organizations to leverage existing models and data in new projects or design variations, leading to substantial savings in upfront design effort. By ensuring that design elements and requirements are represented as machine-readable objects, MBSE supports efficient adaptation and integration of legacy models, reducing redundant activities and minimizing rework. This aspect of MBSE is particularly beneficial in industries with multiple product variants, such as commercial aircraft and automobiles, where reusing proven design components can streamline development processes and enhance overall efficiency [39][36][17][42][47][46].

3. **User Experience:**

- **Better support for automation:** MBSE enhances support for automation by integrating workflow processes into an automated environment, thereby reducing development cycle times. The use of automated simulation allows engineering teams to rapidly evaluate design alternatives and assess their performance, leading to more reliable and higher-quality products. Automated scripts embedded in the MBSE framework can continuously verify the accuracy and consistency of the information within the models, eliminating the need for manual verification traditionally associated with document-based approaches. This automation streamlines the development process, as information is managed and analyzed within a cohesive model rather than fragmented across multiple documents [41][48][47][46].
- **Improved system understanding:** Inconsistencies across documents and among various teams can create significant challenges in maintaining a unified

understanding of the system's design. Teams often work in parallel and update documents at different rates, resulting in a lack of cohesion, which leads to confusion and a fragmented view of the evolving system, making it difficult to achieve a shared understanding among all stakeholders. In contrast to these problems with the traditional approach using spreadsheets, which requires users to infer relationships and details from textual descriptions and cell locations, MBSE provides a structured and explicit representation of system knowledge with functional decomposition, allocation of functions and control flows, offering a clear and consistent depiction of the system's fundamental properties, enhancing overall system understanding and alignment across the development team [21][33].

- **Reduce effort / burden of SE tasks:** The adoption of MBSE reduces the cognitive effort and burden associated with systems engineering tasks. Traditional systems engineering often requires extensive manual handling of tasks related to modeling, simulation and verification, which can lead to high cognitive load and potential for errors. MBSE addresses this challenge by automating many of these tasks, leveraging modeling and simulation capabilities that are directly integrated into the model [48].

  The reduction in manual processing and the automation of high-complexity tasks, such as those related to model verification, decrease the cognitive load on engineers, leading to improved performance, decision-making and overall efficiency. This is particularly evident in the automation of checking processes, where MBSE has been shown to decrease the mental effort required, thus allowing engineers to focus more on higher-level cognitive tasks rather than repetitive or error-prone manual activities. As a result, MBSE enables more effective management of complex technical challenges and enhances the efficiency of engineering processes, making it a valuable tool for reducing the overall burden and improving the life of systems engineers [48][46].

4. **Knowledge Transfer:**

   - **Improved architecture:** MBSE improves architectural understanding by capturing a more complete and detailed representation of architectural knowledge compared to a traditional document-based approach. This comprehensive capture reduces the risk of information loss or mistranslation inherent in document-based systems. Additionally, well-architected MBSE models enable sophisticated queries and automated report generation, providing deeper insights and generating multiple views that manual diagrams alone cannot offer [41][49][33].
   - **Improved communication and collaboration:** One of the most frequently cited benefits of MBSE, and a key aspect for which it is widely recognized, is its ability to enhance communication and collaboration. Through the use of unified, machine-readable models, MBSE ensures that all stakeholders have

access to consistent and up-to-date information, thus improving understanding of requirements and their dependencies. This facilitates effective feedback loops, reduces the risk of omissions, and supports earlier design change decisions. The dynamic and visual nature of MBSE models support better coordination across various design teams and with external partners, making it a crucial tool for efficient and collaborative development [39][49][41][36][17][42][45][50][46].

- **Enhanced knowledge capture / transfer:** The system model can be used to produce a Mission Concept Report and Project Requirements Document (PRD) for the Mission Concept Review (MCR). The model-based approach allows for the creation of different views tailored to specific needs, such as conceptual views for capturing architectural aspects and realizational views for specific implementation requirements [44].

  These model-generated products / documents for the decision gates are particularly valuable because they can be quickly reorganized or updated through query adjustments rather than altering the document itself. This iterative process facilitates the rapid incorporation of changes and ensures that all information remains consistent and up-to-date. Consequently, MBSE provides a more accurate, multidimensional representation of system design information, such as requirements, design rationale and interrelationships, compared to document-based systems engineering, thus enhancing both the capture and transfer of critical knowledge [39][41][44][33].

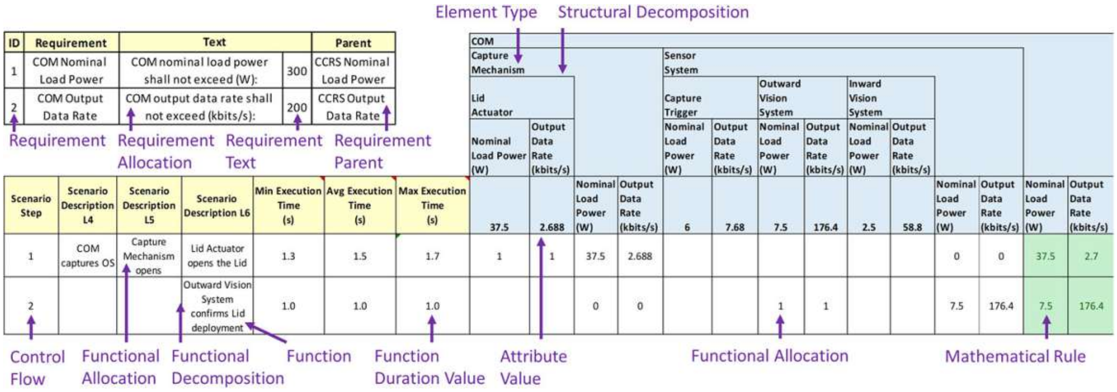
While this list outlined many of the various benefits of MBSE based on the literature, we believe seeing an example from practice can provide a clearer understanding of their real-world impact, even though the provided images / models can be difficult to fully comprehend without any further knowledge in modeling or engineering.

3.1.1 Example from practice

Up until now, we have extensively discussed various aspects of SE and potential improvements that MBSE aims to bring to the engineering world. However, it can be challenging to convey the actual differences MBSE brings to the product / system development life cycle without a hands-on demonstration using a modeling tool to show the system model and different views. The complexity lies in the multitude of views (or diagrams) and all the interwoven aspects that come with it, like automation, or verification and validation, which are all part of the real value of MBSE.

Despite this challenge, providing a real-world example is essential for readers to better understand the practical application and benefits of the model-based approach. We chose an example from [48], which compared a Non-MBSE and an MBSE approach in developing an orbiting sample Capture and Orient Module (COM) architecture for a potential Mars Sample Return (MSR) mission. This example covers multiple aspects that effectively illustrate advantages of MBSE. It highlights several previously discussed benefits and values of the model-based approach in action. While this example is extensive for the purposes of this thesis, it represents only a small fragment of the overall MBSE solution and its contributions to improved results and successful mission / project outcomes throughout the product life cycle.

Requirements are an essential concept of SE, they ensure the finalized product meets the stakeholder needs. The amount of requirements in a complex SoS can rise very quickly and become difficult to manage. The following figure shows an example with just two requirements, but the reality can be closer to hundreds or thousands of requirements [51].



ID	Requirement	Text		Parent
1	COM Nominal Load Power	COM nominal load power shall not exceed (W):	300	CCRS Nominal Load Power
2	COM Output Data Rate	COM output data rate shall not exceed (kbits/s):	200	CCRS Output Data Rate

Scenario Step	Scenario Description L4	Scenario Description L5	Scenario Description L6	Min Execution Time (s)	Avg Execution Time (s)	Max Execution Time (s)	COM													
							Capture Mechanism				Sensor System									
							Lid Actuator				Capture Trigger		Outward Vision System		Inward Vision System					
							Nominal Load Power (W)	Output Data Rate (kbits/s)	Nominal Load Power (W)	Output Data Rate (kbits/s)	Nominal Load Power (W)	Output Data Rate (kbits/s)	Nominal Load Power (W)	Output Data Rate (kbits/s)	Nominal Load Power (W)	Output Data Rate (kbits/s)	Nominal Load Power (W)	Output Data Rate (kbits/s)	Nominal Load Power (W)	Output Data Rate (kbits/s)
							37.5	2.688			6	7.68	7.5	176.4	2.5	58.8				
1	COM captures OS	Capture Mechanism opens	Lid Actuator opens the Lid	1.3	1.5	1.7	1	1	37.5	2.688							0	0	37.5	2.7
2			Outward Vision System confirms Lid deployment	1.0	1.0	1.0			0	0			1	1			7.5	176.4	7.5	176.4

Figure 3.1: Example of Non-MBSE computerized model [48]

First, we look at the Non-MBSE approach. In the top-left of Figure 3.1, we see two requirements. The given values of these requirements have to be checked during development in order to verify that the COM meets both of them. How this is done with the traditional approach, can be seen on the rest of the picture. A computerized model of the COM system has to be manually programmed in a spreadsheet using Microsoft Excel. This spreadsheet gathers the necessary system knowledge from slides, manuscripts and other spreadsheets of the conceptual model and integrates them in the new one. The scenario descriptions on the left part show the hierarchical levels of the system that have to be gone through to find all the relevant parts that can affect the nominal load power and output data rate. On the right side, these parts are displayed in structural hierarchy of the system, with lower level elements being in the smaller rectangular boxes within the higher level ones. The structurally highest one is the COM, where a mathematical rule has to be put in place, to calculate the final result based on the input cells from the other elements [48].

All of these inputs of the Non-MBSE approach were entered manually and the verification that everything was captured and translated correctly into the new spreadsheet was also done manually [48]. For the verification of the COM a time series chart depicted in Figure 3.2 shows the calculated results of the two requirements that were built using spreadsheets for the experiment scenarios of the simulation.

The same outcome had to be produced with the MBSE approach, which used SysML

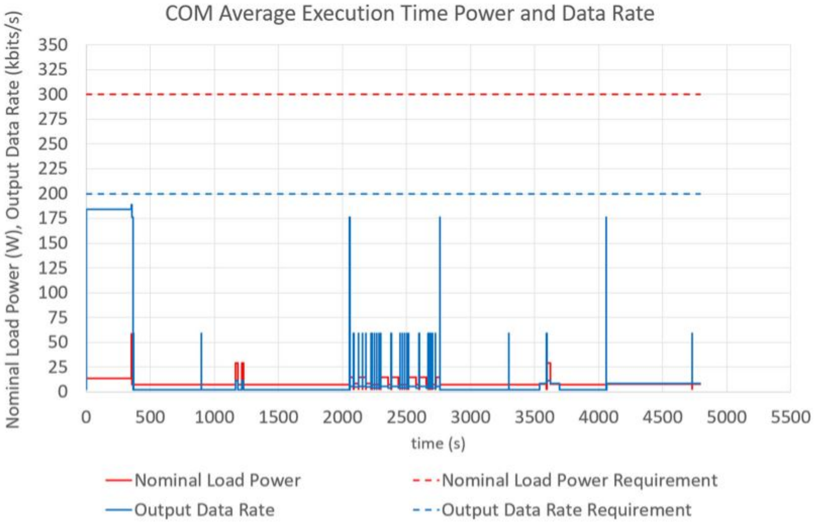
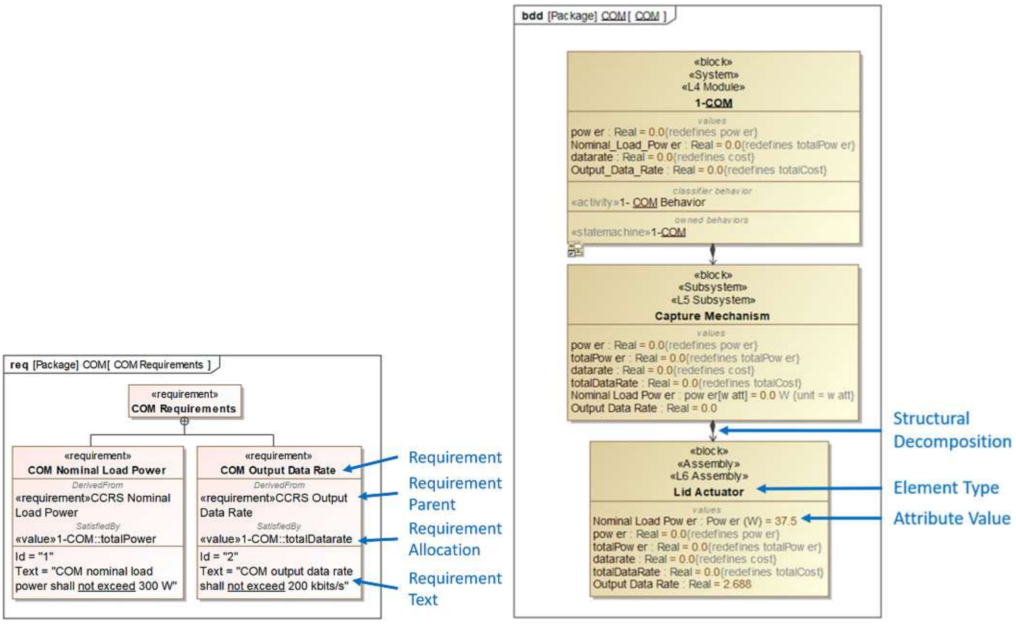


Figure 3.2: COM Non-MBSE approach time series chart [48]

as modeling language and Cameo Systems Modeler as modeling tool. The requirements were captured in a requirement diagram, see Figure 3.3a. A block definition diagram was used to capture attribute values within the block specification, which can be seen in Figure 3.3b.



(a) MBSE requirement diagram [48] (b) MBSE block definition diagram [48]

Figure 3.3: COM Example

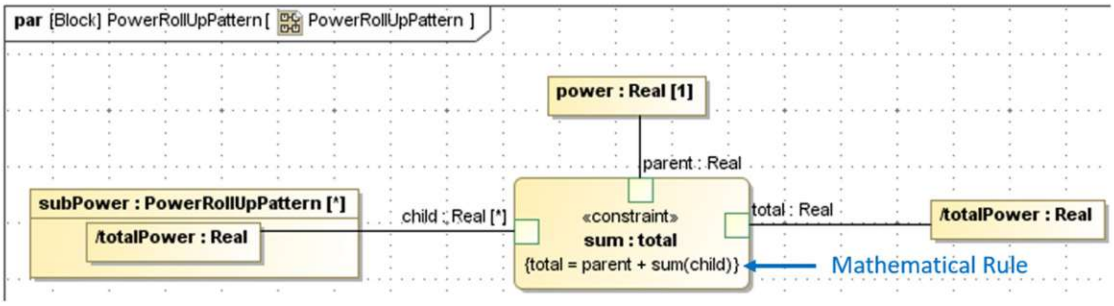


Figure 3.4: MBSE parametric diagram [48]

These values can be accessed and were used to apply the mathematical rule for both power and data rate attributes, which can be seen in Figure 3.4, where the total power is calculated by adding the power from the parent to the sum of all the children, similar to what was done in the excel spreadsheet for the traditional approach [48].

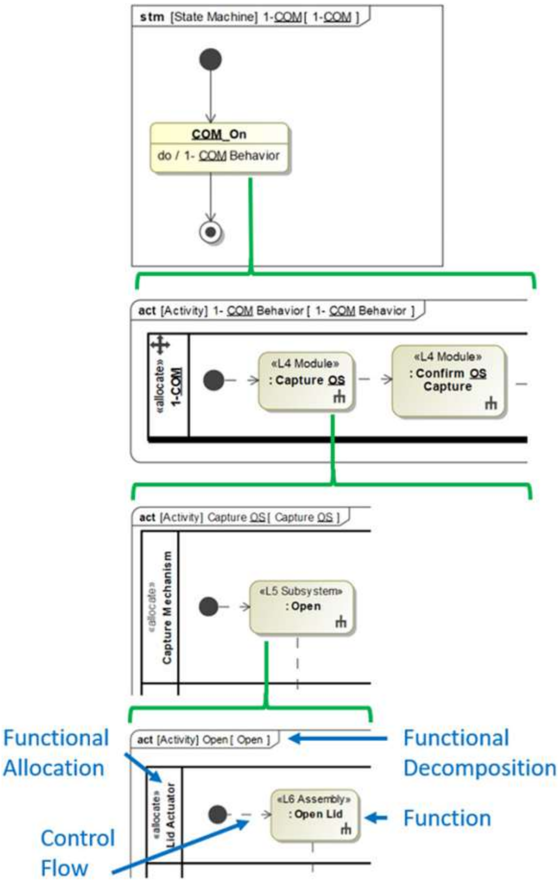


Figure 3.5: COM MBSE behavior state machine and activity diagrams [48]

If it was not apparent from the Non-MBSE approach yet, this simulation example checks that both requirements are fulfilled, specifically during the "Open Lid" activity of the COM, including all activated subsystem elements (reason for the sum of all children in the mathematical rule). This whole functional decomposition, where this functionality is captured and what happens before, simultaneously or afterwards, becomes apparent when looking at the model views. These diagrams are not specifically constructed for the simulation example. They are the integral part of the whole MBSE approach. Figure 3.5 shows the decomposition clearly.

When the COM enters the state ON (is turned ON), a specific behavior begins, which is depicted in the first activity diagram below the initial state machine diagram. There, *Capture OS* is the first activity happening. The little fork on the bottom right of the activity indicates, that there is another diagram describing the activities happening during this activity, in the structural lower level. There, *Open* is the first activity that happens when *Cap-*

ture OS is started. Same thing again, in the next lower level, the activity *Open Lid* happens as the first step.

Figure 3.7 shows the activity diagram of the activity *Open Lid*. It is not necessary to understand the specific steps here, but it is kept as a reference for how a specific model view looks like in practice. Hopefully this decomposition shows how everything is connected in MBSE. When working in Cameo Systems Modeler, every step can easily be walked through by clicking on the activity, which opens the next lower level, contributing to an improved understanding of the whole system and managing complexity [48].

In order to end up with the time series chart in the MBSE approach, a simulation and time series chart configuration has to be defined. This happens manually in a simulation configuration diagram, but these configuration specifics also have to be defined in excel for the traditional approach. Other than that nothing has to be added to get from the conceptual model to the computerized one due to Cameo System Modeler's ability to interpret and execute the SysML diagrams.

The mathematical rule defined in Figure 3.4 does not refer to specific values, it just defines where these values can be found. During execution, the values for power and data rate of all activated system elements are gathered and used for the calculation according to the defined rule [48].

The room for error is reduced in comparison to the Non-MBSE approach, as nothing has to be captured and translated by hand, where typos or other human errors, like using the wrong value or the wrong spreadsheet, can happen. Additionally, during the creation of the diagrams the modeling tool complains automatically if a specific rule of the modeling language is violated, e.g. a connection is missing or invalid etc., which further reduces the potential of mistakes compared to a textual approach.

Another improvement with MBSE in this case is that, if the activity diagrams e.g. the activities and possibly the activated system elements change (with this change being represented in the system model), nothing has to be additionally adapted to get the valid verification with the time series chart. The values will automatically come from the new elements. In contrast, for the Non-MBSE approach various documents, spreadsheets etc. will have to be located, analyzed and updated, where the potential for errors and inconsistency is high [48].

Finally, ending up with the two time series charts, the two requirements can be verified. Figure 3.2 shows the chart from the Non-MBSE approach. Here, it has to be manually verified by looking at the dashed requirement lines, where the potential for human error is introduced again. Now, comparing Figure 3.2 with Figure 3.6 from the MBSE approach, we can clearly see a difference. If a requirement fails it is distinctly stated and highlighted automatically in the resulting chart, reducing the chance for error. The errors in this case were created on purpose for the demonstration of failing requirements in the MBSE approach [48].

Overall, this example should demonstrate how MBSE impacts the work happening in

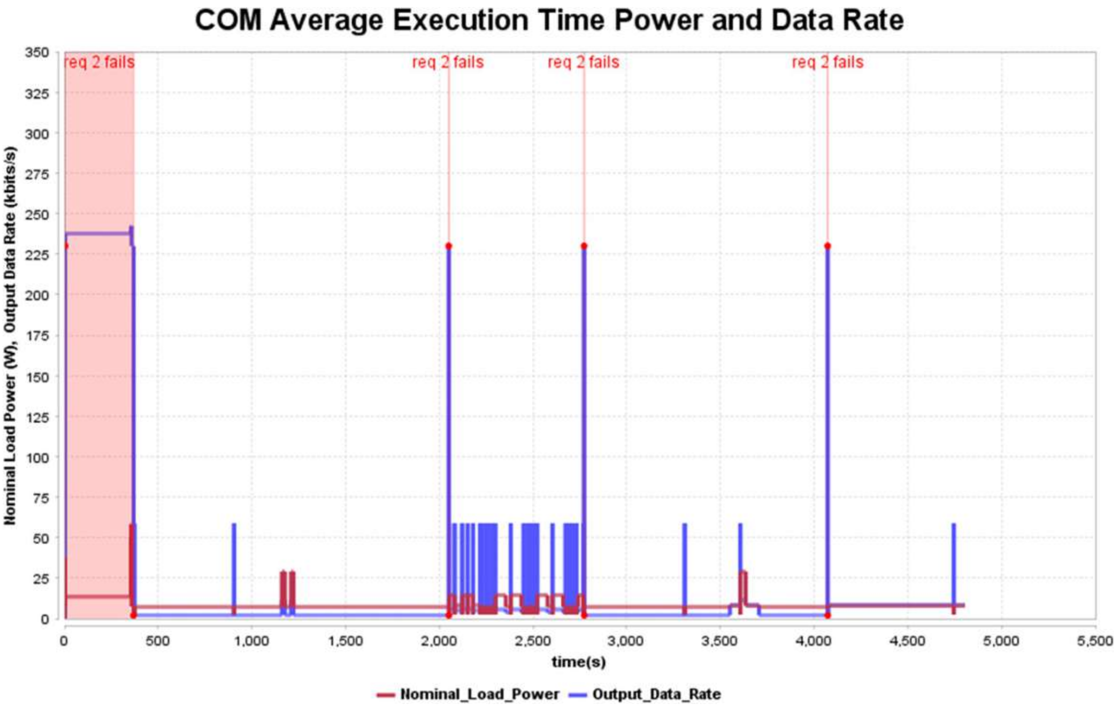


Figure 3.6: COM MBSE approach time series chart [48]

practice, as well as illustrate the mentioned values and benefits from the section before. Additionally, it clearly conveys that MBSE is not a quick and easy solution. MBSE is an approach that takes a lot of effort, especially in the beginning, but the return can be significant.

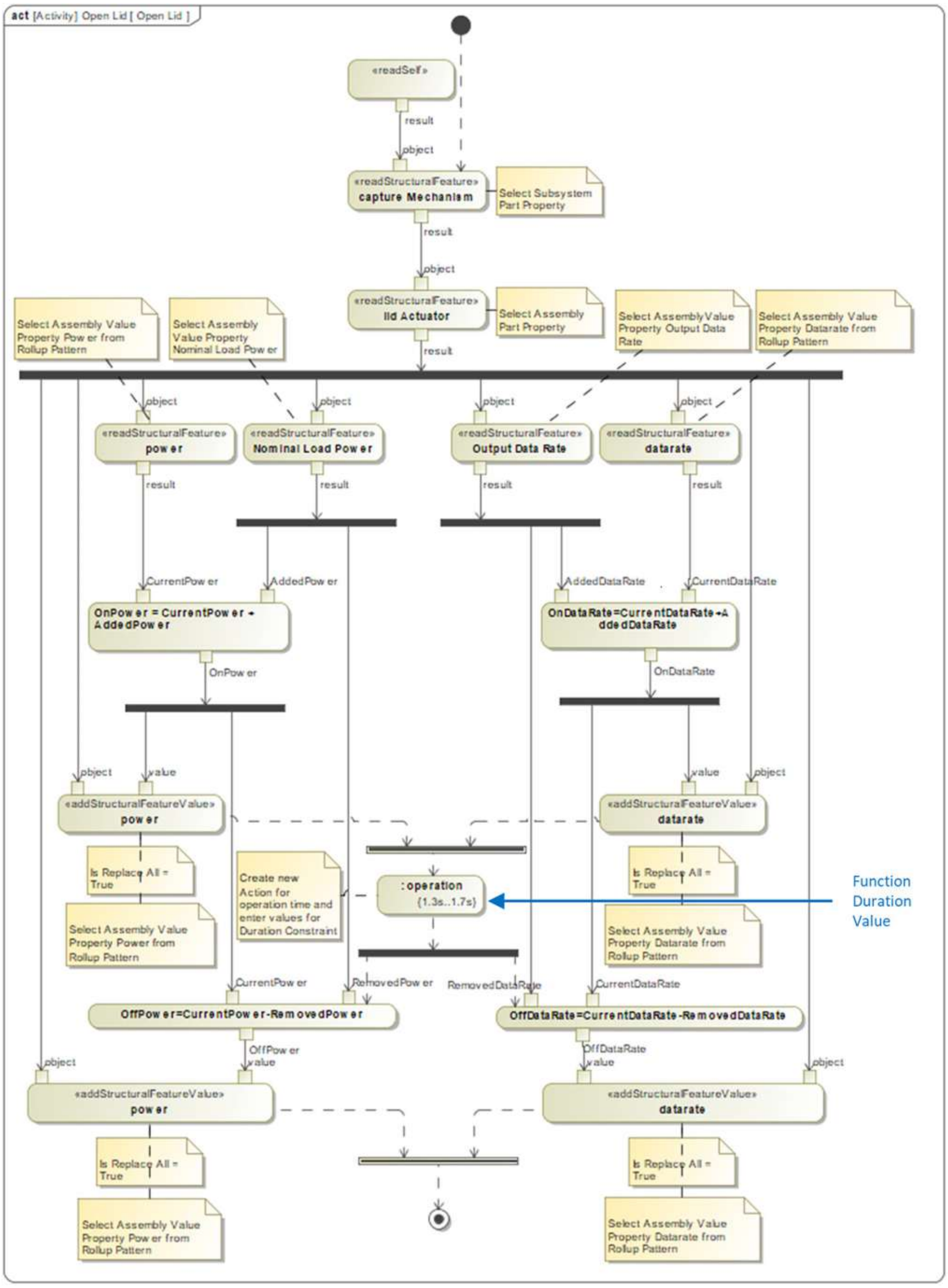


Figure 3.7: MBSE activity diagram of activity *Open Lid* [48]

3.2 Measured Improvements and Return on Investment

Return on investment is a critical consideration in any major business decision. In the previous section, we highlighted the wide-ranging benefits of MBSE as identified across numerous sources, each influencing different stages of the development lifecycle. While these benefits collectively make a strong case for transitioning from traditional SE to MBSE, as illustrated by the practical example, this shift requires careful deliberation. MBSE demands significant investment in time, resources and sustained effort for successful implementation.

For leadership to commit to such a transformative decision, they need compelling evidence and robust justifications to accept the associated risks. Providing this kind of evidence has been shown to be quite difficult for MBSE. [52] reports that although many benefits are mentioned throughout various sources they vary depending on the implementation. The spread across benefit categories is very wide and there was no commonly accepted evidence of benefits found in the 360 papers they looked at. Consequently, management cannot count on a guaranteed, quantifiable benefit when transitioning to MBSE, instead, much of the success depends on how effectively the transition is planned and executed.

Similarly, [53] gathered 2900 claims about MBSE from 60 academic sources. They looked at both positive and negative statements regarding MBSE, where the positive ones align with the benefits mentioned in the section before and the negative ones are dominated by costs, time (to transition) and working hours. They conclude that throughout the literature the mentioned benefits highly outweigh the drawbacks in a 6:1 ratio, which sounds assuring, but there is a caveat that is also mentioned by [52]. Many of the cited benefits are perceived values which are qualitatively measured. They are claims made based on the perception of improvement of people involved in the MBSE transition. Very few papers include quantitatively measured benefits, which makes it difficult to rely on those stated benefits when trying to make an educated decision. Even though the papers that do include metrics report heavily positive towards MBSE [53], the limited availability of numerical data raises uncertainty about the strength of the justifications for these benefits.

The difficulty in measuring benefits is inherent in large-scale transitions like MBSE, where actual returns may take years to materialize. This challenge, has been widely recognized in current literature [54][52][53][55] where the call for programs quantifying and comparing their processes has been placed and is also addressed as best practice in this thesis.

Nevertheless, literature including measured improvements do exist and can help provide economic justification for the transition to MBSE. An early literature review from 2016 [39] reports on multiple projects stating measurable improvements. The first one compares development costs, where an MBSE approach achieved 55% less costs that projects using a traditional approach and on-time delivery being improved from 59% with SE to 62% with MBSE. These improvements are mostly realized by enabling engineers to discover defects early in the project development life cycle, where changes are less expensive and thus avoid rework later on, which would impact both cost and schedule. A different project reports that changes in requirements late in the development lifecycle can lead to 5 or 10 fold increase of SE effort required which would likely cause significant cost overruns. Finding and fixing defects early is critical to stay within cost, schedule and quality objectives [39].

Another project attributes the success of MBSE to the improved design decisions the approach enabled. Major changes to system design often occur after big decision gate reviews as many stakeholders are involved. Using MBSE, a requirement change in a project was discovered that had not been reflected in the design and could be reworked before a design review was conducted which reduced the design iterations necessary. The traceability provided by MBSE ensures that omissions do not happen as a change impact analysis can be performed [39]. This example again illustrates the difficulty to measure the benefits of MBSE, as it did not reduce costs directly, but allowed for more informed design option decisions which improved reliability and prevented costly rework later on.

[39] moreover reports on a project that measured a dramatic 68% reduction in defects after introducing MBSE practices which is attributed to the improved understanding of the system behavior with MBSE, as manual processes using DBSE can produce poor traceability reports and incorrect specifications limiting the understanding of the functional behavior that is needed. They also mention a project where MBSE enabled to do more work with fewer members as well as a large aerospace system development project which estimated its ROI at 40%.

A recently published study evaluated the impact of MBSE adoption on three NASA Mars missions, reporting notable improvements [56]. On one mission, the defect undetected rate dropped from 21% to 12%, while on another, it decreased from 21% to 3%. The study attributes these gains primarily to effective code reuse, functional decomposition of flight systems, abstraction of system elements and other MBSE techniques, highlighting the significant advancements achieved through MBSE [56].

The study used for the detailed example from practice in section 3.1.1 was also able to measure a significant difference between their MBSE and Non-MBSE approach. They report that for the Non-MBSE approach, all 5758 knowledge elements had to be manually processed for the modeling and simulation process, while for the MBSE approach, 2824 of them were automatically processed. This is an improvement of 49% during knowledge processing and underlines the higher level of support for automation with MBSE, which reduces the effort and burden of SE tasks [48].

One of the most detailed and often cited projects regarding improvements of MBSE, which also include calculations for their ROI, is the Submarine Warfare Federated Tactical Systems (SWFTS) [57][51] program, which transitioned from traditional document-centric systems engineering to a model-based systems engineering approach and focused on clear evidence of improvement during their change. The involved system is a rapidly evolving combat SoS, which has to adhere to annual baselines with full system integration and certification processes as it has to be ensured that the combat system installed on the ships is operationally suitable, effective and interoperable. Simultaneously, there are many sources of new requirements, such as new or revised capabilities, cybersecurity aspects, operational challenges reported from current fleets, new technologies, etc.

The baselines were manually constructed and managed in large spreadsheets with hundreds of columns and thousands of rows documenting interface requirements, data methods and which component systems provided or required the information defined by the requirement / method (similar to the example from practice given in section 3.1.1). The initiative to introduce MBSE to SWFTS began in 2009 and involved the conversion of approximately 2700 interface requirements from a document-centric requirements database to a model-based process [51].

During their transition of the legacy systems engineering products they discovered many defects that have crept in over the years as they have been manually maintained. Just the use of the modeling tool of the MBSE approach, which enforces requirements to be internally consistent, allowed the detection of these mistakes [51].

Although building the baseline SE products with the MBSE approach did require more labor hours the higher quality of the model-driven products did convince leadership to transition all subsequent baselines. Over time they climbed the learning curve and were able to improve the average cost of a change request and additionally were able to develop scripts to generate tailored SE artifacts for individual systems that improved efficiency of component system developers, which wouldn't have been cost-effective to do manually [51].

They reduced the average time needed per requirement to 9.9 hours down from 12.1 hours with the legacy SE process, which is a significant change when working with thousands of requirements. They were able to discover 192.3 interface defects per 1000 requirements changed, with the MBSE approach compared to 145.9 with the SE approach. This is an increase of 32% of defect discovery using MBSE, which they attribute to the higher quality of the artifacts produced from the SoS model. Furthermore, the MBSE approach allowed them to reduce the discovered errors during platform testing by 37% from 212.1 interface defects with the legacy SE approach down to 132.9 with MBSE. Although the first 32% increase in defects discovered might seem troubling, they found that using their new approach, 18% of interface defects are discovered earlier in the life cycle, where defect eradication has been proven to be significantly cheaper as errors discovered during platform testing cost between four to fifty times as much to correct than compared to during integration [51].

Overall their new approach resulted in a 9% reduction in total interface defects identified. Although MBSE was not the only change implemented, the research strongly suggests that MBSE was the primary factor behind this reduction. This improvement is a key driver of ROI for MBSE, as defects often represent a significant portion of development costs [51]. Although defect removal costs vary by project and defect type, any engineering approach that reduces defect introduction or enables earlier defect detection will lead to significant savings in development costs.

The SWFTS program reports savings between \$3.65 million (where only the reduction in up-front SE labor was included) and \$25.6 million (where the reduction in interface requirements baseline change and the shift of defects to an earlier stage was additionally included) in the first 5 years after transition. This results in an estimated ROI between 1.1 and 7.75 over 5 years of their \$3.3 million transition investment. Apart from these estimates they report on benefits resulting from the MBSE transition including centralized documentation, managing increased system complexity, automated data validation and enhanced traceability between capabilities, requirements, functions and deployment, which cannot be accurately quantified and are therefore not included in their ROI estimates. They further indicate that the observed ROI should be considered a lower bound, as process efficiencies could still be enhanced. Additionally, their steady expansion in scope despite a constant budget suggests that the benefits of the MBSE transition continue to accumulate [51].

3.2.1 ROI across Domains

The examples given above provide clear, measured evidence of the improvements enabled by MBSE, but are outnumbered by cases reporting qualitatively assessed benefits. Even fewer studies offer specific ROI estimates based on concrete, measured improvements. This scarcity of quantitative ROI evidence likely stems from the extended time period required for MBSE transitions in development processes or programs, as well as for the ROI to materialize and become estimable. For example, the large-scale SoS transition within SWFTS began in 2009, with detailed reports on the transition's progress appearing in 2014 [57] and final reports showing measured improvements and ROI published only in 2021 [51]. This 12-year span from the first steps of the initiative to ROI reporting underlines the difficulty in securing leadership buy-in for MBSE transitions, who often rely not only on justification based on engineering feedback, but specifically on economic incentives.

Figure 3.8 given by [17] shows the development of cost over time with traditional SE and MBSE, including factors that contribute to the investments and gains. Clearly, there is a significant gap between the costs in the early stages, but over the long run, with all the various benefits accumulating, elements being used and reused throughout projects, the economic justification becomes apparent.

Conclusive with what is shown in Figure 3.8, [39] argues that a project that does not use an MBSE approach throughout the full system development life cycle, will likely have all the investment costs, but won't be able to reap the benefits or ROI hoped for. Using MBSE only for requirements management for example, might already achieve marginal benefits, but the significant improvements reported qualitatively and quantitatively in the past sections won't be possible without a coherent approach spanning multiple life cycle phases and stakeholders [39]. The initial progress with MBSE requires the most amount of time and learning, as neither experience nor reuse libraries are available. Over time, added elements can be reused and repurposed and the required time for systems and components will decrease and improve the return provided by MBSE [58].

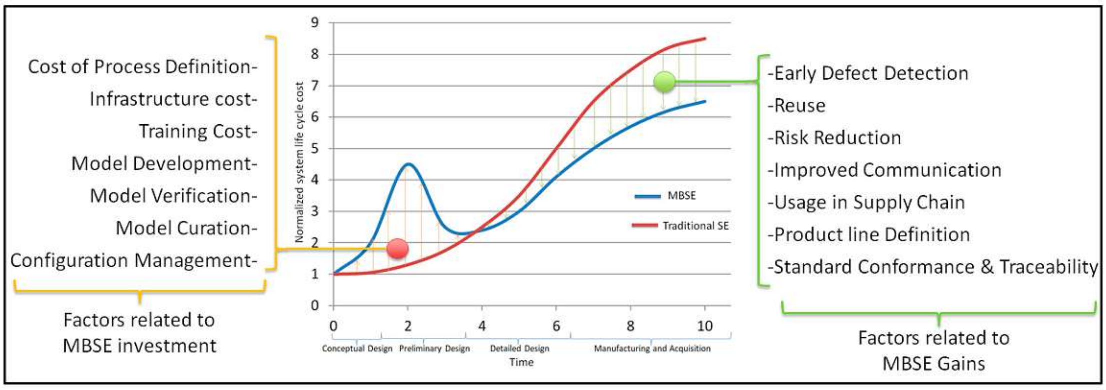


Figure 3.8: MBSE investments and gains compared to traditional SE [17]

The specific benefits perceived or measured by MBSE vary greatly and depend on the implementation [52] as well as the domain of the project [59]. Consequently, so does the potential ROI of an MBSE endeavor. [17] argues that MBSE investment and expected gains largely depend on the system's intrinsic characteristics, which they define as [17]:

1. **Complexity:** is considered to be a function of the number of unique components of the system, the interactions between them, the amount of knowledge required for developing the system and the amount of knowledge required to describe the system.
2. **Operational Environment:** characterized by the number of stakeholders, the external systems that mandate regulatory and interface constraints as well as the applicable standards that the system needs to conform to.
3. **System's lifespan:** can be characterized by the system's useful life, meaning all life cycle stages, which were covered in section 2.2.2.

Their argument is that different industries / domains can benefit to various degrees from MBSE depending on those three factors. As examples for system complexity they mention aerospace systems such as airliners with more than 100.000 parts (high), high-tech industry systems like copying machines with around 2000 parts (medium) and systems of household consumers used on a daily basis that have around 300 parts (low). Similarly, system lifespans vary widely across industries, with relatively short ones from 0 to 1 year compared to systems with a long lifespan of more than 30 years [17].

The final conclusion of this analysis, is shown in Figure 3.9 of [17], revealing that transportation and mobility, aerospace and defense, energy, process and utilities, and natural resources sectors (e.g. mining, oil, gas, agriculture, forestry) are expected to gain the most from MBSE adoption. Due to increasing system complexity, marine and offshore, architecture and construction, life sciences, and industrial equipment industries may also benefit significantly over time. Conversely, sectors such as financial and business services, high-tech and even less lifestyle and fashion as well as consumer packaged goods are anticipated to gain comparatively less from MBSE adoption. The connection between system complexity and MBSE adoption potential is further supported by findings in [60], which align with those of [17].

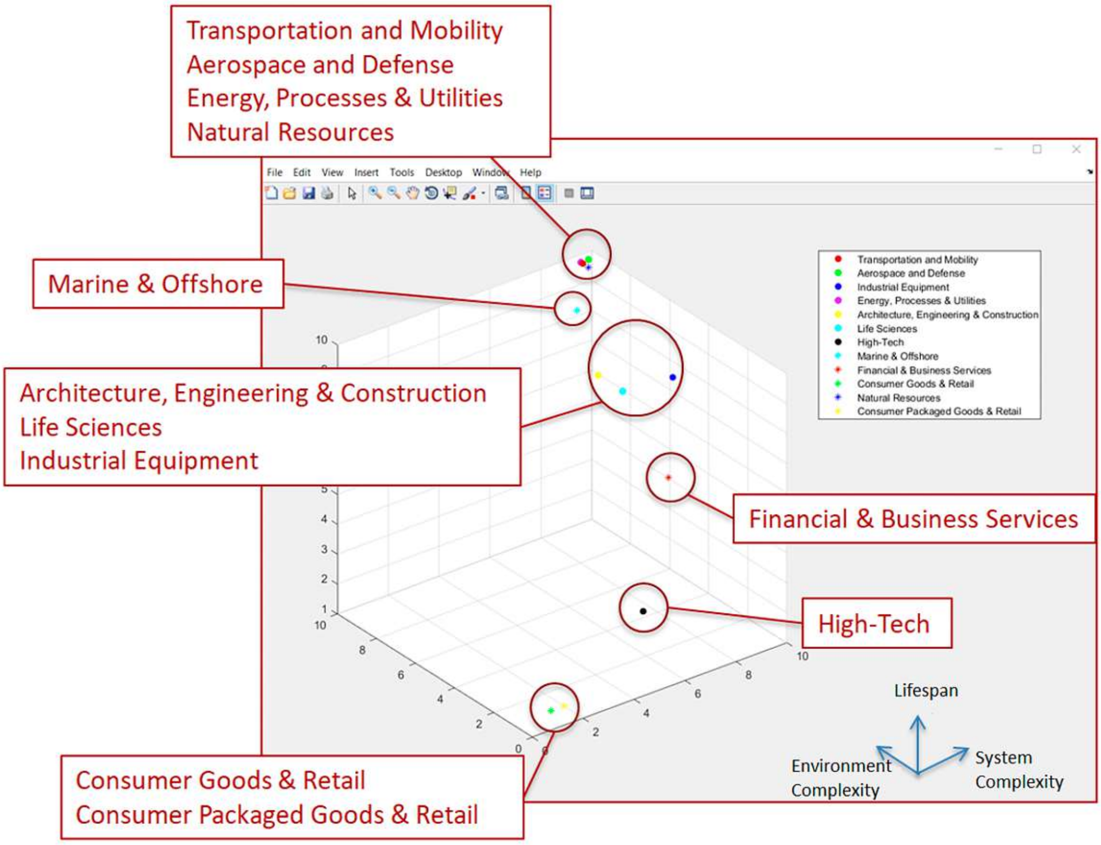


Figure 3.9: MBSE potential value across industry sectors [17]

Regardless of the domain, adopting MBSE requires substantial adjustments within an organization as it reshapes existing processes and working habits. As outlined, MBSE yields its full value when applied across the entire system life cycle rather than within isolated processes, helping to justify its significant upfront investment. However, determining the right starting point and prioritizing tasks, even with a chosen modeling methodology, can be challenging. Based on a meta-synthesis of extensive research, this thesis provides a maturity assessment to help organizations identify their current stage in the MBSE adoption process. This assessment is designed to guide companies in focusing their attention and resources on key areas shown to be critical in other successful MBSE implementations, helping to maximize both efficiency and impact as they progress. The following section will explain the methodological approach of our research and the development of the maturity assessment in detail.

CHAPTER 4

Methodological Approach

The main reason we decided on a qualitative meta-synthesis was the desire to comprehensively analyze and synthesize the body of evidence on the transition to model-based systems engineering, to achieve robust and broad conclusions and implications. Similar to MBSE the statement *"The whole is far greater than the sum of its parts"* can also be said about reviewing various scientific sources. Individual studies can never be definitive, bringing together the results of many different individual studies, synthesizing and evaluating them, and uncovering consistency far extends what any single study can ever achieve [61]. Consequently, we follow and extend the research methodology of a meta-synthesis defined by [62][61] which consists of a Systematic Literature Review [63], the full-text review and the synthesis of the findings of the final amount of relevant papers. All of the included phases will be discussed in the remainder of this chapter.

4.1 Systematic Literature Review

A literature review is an important part of any scientific work. Gathering knowledge and resources is one of the first steps of conducting scientific research. During this step it has to be evaluated whether a gap in current research exists. This process of familiarizing oneself with a certain area of interest most commonly happens in a non-systematic way, where different libraries are queried with various inputs in order to gather a broad understanding of the current knowledge base on the topic. After this preliminary part the actual first phase begins.

Phase 1: Definition of the research scope

Given the breadth and depth of resources identified during the initial literature research on the adoption of MBSE, encompassing various aspects, challenges and best practices, we realized a gap of a holistic perspective on the area of transition to MBSE. Therefore the decision was made to dive deeper into this subject. Which involved focusing our attention on addressing the research questions outlined in the introduction 1.1 as well as the conclusion 7.

The definition of these research questions and therefore the scope of the study is essential to be able to start the next phase.

Phase 2: Conduct systematic search

The most common search strategy when conducting a literature search is a keyword search in various electronic databases [64]. However, this may result in an overload on potentially important studies, as online databases can return hundreds or thousands of papers. Illustrated by the example of the keyword *MBSE*, which has 4115 hits on Scopus [65] and 672 on IEEE [66]. Therefore, in order to limit the amount of relevant papers and more precisely only find those that cover some aspect important to the defined scope, the decision was made to develop a precise search query using Boolean operators. After several iterations of adding, due to additional relevant hits and deleting, due to basically breaking the query resulting in either no hits at all or way too many, where either case is undesirable, the finally resulting search query was the following:

$$Q = (\vee \mathrm{M}_i) \wedge (\vee \mathrm{A}_j)\,, \ \text{where}$$

$$\mathrm{M}_i \in \left\{ \begin{array}{l} \text{"mbse" OR "model-based system* engineering" OR "document-cent*} \\ \text{system* engineering" OR "dbse" OR "digital engineering" OR} \\ \text{"digital model-based engineering"} \end{array} \right\}$$

$$\text{, and}$$

$$\mathrm{A}_j \in \left\{ \begin{array}{l} \text{"adoption" OR "implementation" OR "transition" OR "switch*" OR} \\ \text{"experience*" OR "challenge*" OR "strateg*" OR "best practice*"} \end{array} \right\}.$$

In spoken terms the query describes that we are looking for resources that include the term *mbse* or any of the other listed labels in the first group as those are often used synonymous in research. A study is only relevant though, if it also includes one of the terms of the second group. This way, we filter out those papers that just engage with MBSE in some form, but do not regard the process of adoption. The asterisks in the query are used as placeholder for any number of unknown characters. This can be utilized to specify the plural form of the given term, but also to search for different endings of a term, e.g. strategy - strategies - strategic or centric - centered. This enables us to be more inclusive in our search query, while still filtering unnecessary overload. Following the advice from [64] we decided to use this query on multiple electronic databases. While initially the three well respected scientific databases *ACM Digital Library, IEEE, Scopus* were decided upon, we additionally chose to include the *Wiley Online Library*, as it specifically offers access to papers from the INCOSE (International Council on Systems Engineering), which already in the initial keyword search turned out to be very helpful and comprehensible resources and often times were not covered by the other three libraries.

Phase 3: Screening of papers

After gathering a notable amount of papers through the use of the query on title, abstract and keywords on the four mentioned electronic libraries, it was necessary to filter only those that are essential to answering the research questions. Therefore we defined several inclusion and exclusion criteria.

Inclusion criteria:

IC-1: Publications reporting on both MBSE and the adoption / implementation / strategies used (as defined by the search query)

IC-2: Peer-reviewed publications (i.e., articles in journals, conferences, book chapters)

Exclusion criteria:

EC-1: Publications considering only MBSE, DBSE or Digital Engineering

EC-2: Publications not written in English

EC-3: Publications where the full text is not accessible

EC-4: Non-Peer-reviewed and non-scientific publications

EC-5: Publications with less than 5 or more than 60 pages

EC-6: Publications older than 10 years (before 2014)

The reasoning behind most of the criteria is simple as they are standard inclusion / exclusion criteria for scientific work written in English. The decision to omit works published before 2014 was primarily driven by the substantial volume of relevant research already identified within the past decade. Our preliminary findings indicated that the most significant developments and comprehensive studies on MBSE adoption have occurred in the last ten years. Including literature prior to 2014 would have introduced additional effort with only marginal returns. Given that more recent publications typically encompass and build upon earlier findings, this solution still ensures that our synthesis captures the evolution of thought and practice in MBSE without the need to re-examine older sources. This approach allows us to focus our efforts on the most current and impactful studies, thereby enhancing the quality and relevance of our meta-synthesis.

The search was performed on 5^{th} of March 2024 on all four databases and yielded a total number of 1672 potentially relevant publications. For all of these hits the BibTeX

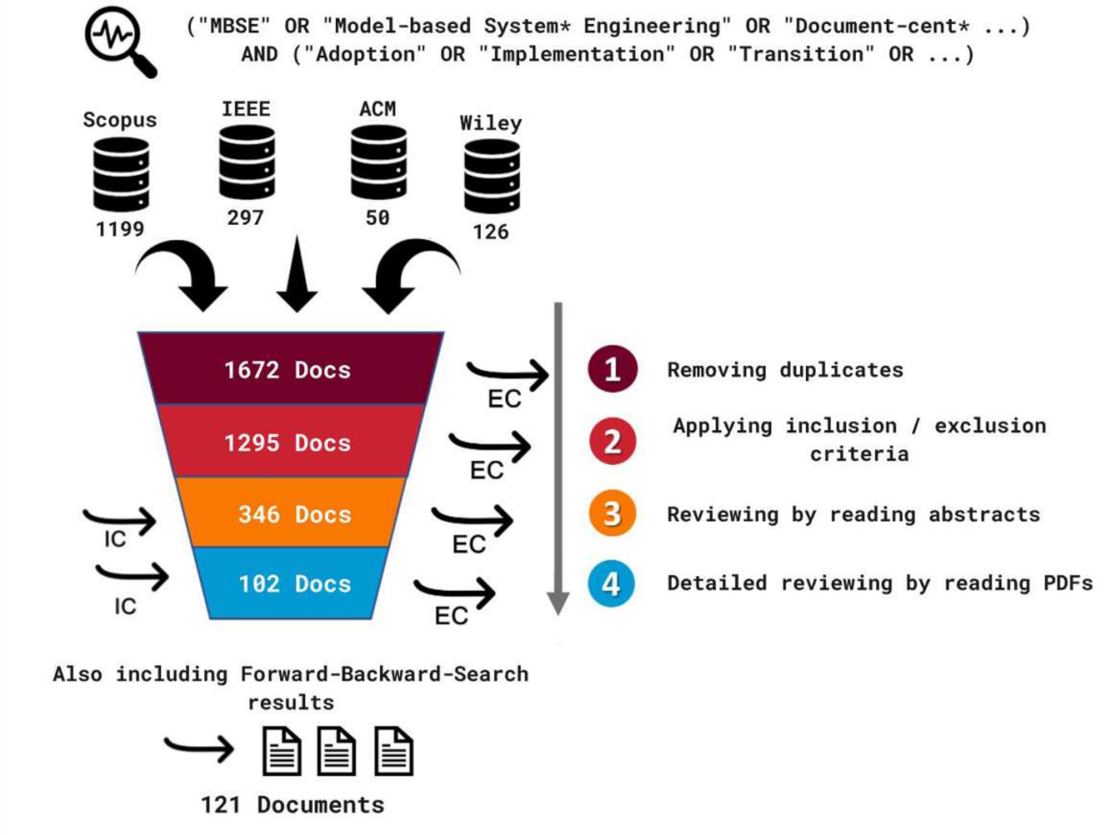


Figure 4.1: Document collection and filtering methods

citations were manually downloaded and put into one large BibTeX file. A python program, provided by my supervisor, was then used to run through all of these citations and formatted the included information into a CSV document while simultaneously detecting and removing duplicates based on DOI and Title. After this process step the resulting amount of publications were 1295. Although this is a considerable amount of studies to work through, this approach was chosen to make sure we gain a holistic perspective on the topic and do not automatically sort out publications before further consideration.

Phase 4: Filtering of Abstracts

In the fourth step the python program was used once again to semi-automatically run through the abstracts which were included in the citations. After reading each abstract the source was rated either *Yes*, *No*, or *Maybe* based on how relevant the covered topic seemed. By adding comments to every entry we simultaneously used this process to broadly categorize what topic or sub-area of the topic was written about. After applying this method we were able to reduce the relevant publications (voted with YES) down to 102 papers, while also having 244 additional ones as back-up if some particular area needed further input or explanation.

As a final step, before starting with the Full-Text Review Process we downloaded all of the publications, except for seven as we did not have access to them. To further ensure a comprehensive collection of relevant papers, we decided to conduct both forward and backward searches. The backward search involved examining the reference sections of all downloaded publications, while the forward search utilized Google Scholar's 'cited by' feature to identify subsequent studies that referenced the initial papers. This dual approach is recommended for meta-analysts by [67] to ensure that no relevant studies are overlooked. Additionally, this strategy serves as a countermeasure to publication bias.

Publication bias, also known as the "file drawer problem" or "bias against the null hypothesis", refers to the tendency to publish only research with statistically significant results. This occurs because significant findings are often considered more important, while non-significant results are deemed trivial. As a result, studies with non-significant outcomes are less likely to be submitted or accepted for publication, leading to an overrepresentation of positive results in the literature [61]. This bias can distort the findings of systematic reviews and meta-analyses by inflating the perceived effect sizes and leading to inappropriate conclusions. To counteract this, it is essential to include both published and unpublished research that meets relevant inclusion criteria. Google Scholar, which is reasonably effective in locating gray literature [61], was used in this phase precisely for this reason. Applying this method enabled us to identify 19 additional publications that were deemed relevant to our research questions.

4.2 Full-Text Review Process

After having downloaded the final amount of publications also including the ones identified from the chosen Forward-Backward-Search approach, it was time to start the full text review process. In order to have a better organization and options available considering the amount of research we had to work through, we decided on using the reference management software *Zotero* [68].

The comprehensive review of the gathered sources involved a systematic and thorough reading of the gathered publications. Initially, the sources were categorized into two groups: those identified through the initial search and those obtained via forward-backward citation search. This classification helped to organize the review process efficiently.

The reading sequence primarily followed the chronological order of publication. This approach was chosen, as it facilitates a better understanding of the evolution and acceleration of MBSE adoption over the past decade.

However, the reading order was occasionally adjusted to accommodate related works. For instance, when a paper had follow-up studies e.g. [57][51] or covered very closely connected topics like [30][32][27], these papers were read in succession. This strategy ensured a comprehensive understanding of interconnected research and allowed for a deeper insight into specific topics and the continuity of work by particular researchers or research groups.

By employing this methodical and flexible approach, the review process not only provided a broad overview of MBSE adoption trends but also facilitated an in-depth understanding of significant contributions and developments within the field.

4.3 Meta-Synthesis

Following our systematic review of MBSE transition literature, guided by our defined research questions, we synthesized insights from key studies to construct a cohesive understanding of the current knowledge base. This meta-synthesis, also referred to as qualitative meta-analysis, is not a simple catalog of prior work but an integrative effort to build a new theoretical foundation that advances understanding in this field [61]. As described in [62], the core of meta-synthesis involves extracting and analyzing primary studies to identify common themes and patterns, providing an interpretation of cumulative evidence that enhances theoretical insight and allows for a contextualized contribution.

By synthesizing insights on the challenges, pitfalls, and best practices in MBSE transitions, we constructed a maturity assessment that not only organizes this knowledge but also sets a new baseline in the field. This assessment directly addresses our third research question (RQ-3) by providing structured, high-level guidance for organizations to understand their MBSE transition stage and receive actionable steps tailored to each maturity level. In this way, the maturity assessment serves as an applied synthesis, transforming our gathered knowledge into science-based recommendations drawn from diverse and fragmented findings. This provides practitioners with a structured framework to navigate MBSE adoption with greater clarity and direction.

4.3.1 Maturity Assessment

While developing a maturity assessment fulfills the synthesis requirement of our qualitative meta-synthesis, this approach is atypical for this research methodology. Maturity assessments are generally more aligned with design science research, as they are considered artifacts created to address or improve existing challenges. Consequently, in developing our assessment, we adhere to the research methodology for maturity assessment design outlined by [69] and [70], who build on principles of design science research.

Before beginning development, it is essential to define the purpose of the maturity assessment. Based on the categories defined by [69], descriptive, prescriptive and comparative, our maturity model is prescriptive. Rather than solely assessing the current state, it emphasizes the relationship between domain maturity and business performance, indicating how to approach maturity improvement in order to positively affect business value.

Additionally, this maturity assessment is designed to assist practitioners in navigating the changes required for successful MBSE adoption. While the current literature covers various aspects with regards to MBSE, the challenges and difficulties they encountered, the mistakes they made along the way and the lessons they learned and best practices they discovered, this information remains fragmented across numerous sources. This meta-synthesis, and the resulting maturity assessment, aim to address this fragmentation by offering a structured, comprehensive approach that significantly enhances the resources available for MBSE transition.

In both our initial and systematic literature review, we found no maturity assessment that provides this type of prescriptive guidance, fulfilling the requirement for novelty within the design science research framework [69][70]. The only comparable work we identified are [71] and [20]. The first one presents a descriptive maturity assessment model that assesses an organization's current state and offers benchmarking against competitors or industry averages. While the second one provides no assessment, but valuable decision-support for prioritizing tasks and capabilities during MBSE adoption. [20] introduces a framework used to calculate which capabilities to address first to optimize ROI. While this tool offers a valuable method for prioritizing specific capabilities to optimize ROI during MBSE adoption, it is quite different from our approach. It focuses on task-level decisions, whereas our maturity assessment provides a more holistic view of the entire transition process, addressing challenges, pitfalls, and best practices across all stages. Both papers therefore differ fundamentally from our prescriptive model.

Phase 1: Scope

Determining the scope of the maturity model sets boundaries for its application and use. Our maturity assessment is not focused on a specific domain, but serves as a general guidance for any organization interested in adopting or transitioning to MBSE. It is both meant for academia as a new synthesized baseline of knowledge as well as practitioners on their MBSE journey.

Phase 2: Design

A models is never the true reality, but an abstraction. It is important to find a balance between the often complex reality and model simplicity. Oversimplifying a model might omit valuable information, while keeping it too close to reality may limit interest and create confusion. Additionally, the design of the model has to incorporate the needs of the target audience and how they will be addressed [69].

In our model, the primary motivation for application, **why** users would seek to apply it, is to gain clarity on their current MBSE adoption stage, identify critical next steps and recognize essential actions that may still be pending. **How** it is applied within diverse organizational structures typically involves a self-assessment, conducted by management or cross-functional stakeholders who possess a comprehensive view of the MBSE approach and organizational context. This leads to **who** should be involved: an engineer may lack the organizational insights needed for an accurate assessment, hence the assessment should be done by management, but engineers can contribute valuable perspectives on specific implementation questions. Finally, **what** the model aims to achieve is an informed understanding of the current adoption level, a clear path forward and the opportunity to benefit from the experiences of others, learning from their successes, avoiding common pitfalls and incorporating proven best practices.

A common design principle in maturity assessments is the use of cumulative stages, where higher stages build on the requirements of previous ones. The number of stages may vary across models and should be chosen to align with the specific content and objectives of the assessment [69][70]. We selected four maturity stages to enable clear, logical progression. The label and a compact description of each level can be found in section 5, where we introduce every stage and list the respective elements.

Phase 3: Populate

The next step in the development of a maturity assessment is to identify the information necessary for a deeper understanding of maturity and how this can be measured. The goal is to gather input that, when organized across defined maturity levels, remains mutually exclusive and collectively exhaustive on the topic of interest. For established domains, this is often achieved through a comprehensive literature review [69][70]. In our case, following the decision to structure our maturity assessment into four levels, we used the results of our systematic literature review of challenges, pitfalls, and best practices and systematically assigned each item to the most appropriate maturity level, based on the natural progression of readiness implied by the research. This process ensured a logical and experience-based ordering that aligns with the stages of MBSE maturity.

We found limited guidance in the existing literature on how to prioritize these aspects within maturity levels. While [7] offers some direction by analyzing how contextual factors, such as company size, pre-existance of SE, management support, and industry type, influence best practices, the conclusion remains that all best practices are generally relevant. Their findings confirm that context affects the order of priority rather than the applicability of practices themselves. This insight aligns with our structured approach, as our maturity levels naturally prioritize practices according to the general stage of MBSE readiness. For instance, companies with an established SE foundation may naturally bypass Level 1, allowing them to focus on practices associated with their assessed maturity level.

Similarly, [72] explores the impact of organizational structure on MBSE adoption, noting that factors such as company size, centralization, and vertical differentiation can hinder adoption, whereas formalization, flexibility, and interconnectedness promote it. Although insightful, these findings add specificity that would complicate our maturity model, potentially reducing clarity and accessibility. However, engineering managers might be able to combine this contextual insight with our model to create tailored strategies suited to their organizational circumstances.

As an effort to improve visual clarity and overall structure within each level we used categorization to group elements of the same level and type (challenge, pitfall, best practice), that concern the same subtopic. Additionally, some topics identified in our literature research repeat across challenges, pitfalls, and best practices, reflecting their relevance in different contexts. In the maturity assessment, these elements are interconnected to highlight their close relationships and suggest alternative paths or considerations for progressing through each maturity level. However, in this thesis's results and discussion section, such connections are not displayed, as the primary aim here is to present findings in a clear, organized structure.

Finally, to ensure accurate maturity assignment, we developed four to five targeted questions per level. The self-assessment is structured so that participants start with Level 1 questions and proceed sequentially: if all questions for a level are answered with a *Yes*, they continue to the next level. Once a *No* is encountered, that level is assigned as their current maturity level. This approach balances comprehensive assessment with ease of use, enabling a practical evaluation without an overwhelming number of questions, a consideration that research indicates can enhance engagement and completion rates [69].

Phase 4: Test

Following the phases outlined by [69], this master thesis concludes with Phase 4, the testing phase, where we evaluate our maturity assessment model. Both [69] and [70] emphasize the importance of testing for effective validation. To this end, we implemented an initial validation survey aimed at domain experts. This survey, though brief, includes questions about the respondents' MBSE background, their level of agreement with the assessment, and potential areas for improvement. While this provides an initial validation, a full evaluation would ideally involve extensive case studies to assess the accuracy of maturity assignments and the utility of our recommendations.

Phases 5 (Deploy) and 6 (Maintain) are beyond this thesis's scope. However, it's possible that future work could build upon our contribution to MBSE adoption by using this initial feedback to refine and update the assessment, followed by deeper evaluations through case studies to finalize it. Once fully validated, the assessment could be deployed as a resource for companies seeking structured guidance in transitioning to MBSE. We anticipate that regular updates might be necessary to ensure relevance as MBSE practices and industry needs continue to evolve.

4.4 Validation and Reflexivity

Ensuring validity in this thesis required a rigorous approach to the development, evaluation, and iterative refinement of the maturity assessment model. Guided by a systematic literature review and the principles of meta-synthesis, we gathered a robust set of data. The maturity levels, designed to guide organizations through phased MBSE adoption, were structured to be comprehensive, with cumulative stages that progress logically to address the specific needs of MBSE practitioners.

Given the prescriptive nature of the assessment, a reflexive approach was also essential. Reflexivity helped us remain aware of our own biases and assumptions as we synthesized diverse research findings into the model's framework. This process involved continuously questioning our categorizations and definitions of maturity stages, as well as our choices in prioritizing certain practices over others for each level. To mitigate subjectivity, our decisions were compared with existing literature, and priority was given to data-supported practices from credible studies whenever possible.

For initial validation, we employed an expert survey, recognizing it as an essential first step in assessing the model's relevance and usability. This approach offers preliminary insights into the model's practical applicability and gathers feedback on the assessment's structure and content. While our chosen method provides early validation, we acknowledge that further in-depth testing, such as implementation in case studies, would strengthen the assessment's robustness.

CHAPTER 5

Results and Discussion

In this chapter, we present the structured findings of our systematic literature review, which form the foundation for our maturity assessment. Each of the following sections corresponds to one of the defined maturity levels, each of which addresses a specific phase in an organization's MBSE adoption journey. The assessment categorizes these findings into three groups: challenges, pitfalls, and best practices. Challenges reflect common barriers and difficulties documented across prior MBSE implementations. Pitfalls highlight potential missteps that risk undermining progress or incurring unnecessary costs and resource strain. Best practices provide actionable strategies, distilled from lessons learned across diverse studies, to support and advance maturity at each stage of MBSE adoption.

To provide better structure and clarity in the assessment, we grouped the elements under specific categories that capture their main focus areas. These categories are briefly introduced below and are referenced throughout the following sections of the maturity levels:

- **Knowledge and Skills:** This category encompasses elements related to gaps in knowledge, skills, understanding, or awareness. These issues are often underestimated or assumed to resolve themselves over time. However, research highlights that missing foundational understanding and the time required to develop necessary skills are critical factors in MBSE adoption efforts.
- **Work Culture:** This category addresses cultural resistance to change. Resistance among the workforce is a significant challenge when implementing transformative shifts like MBSE, which fundamentally alter established working habits and processes.
- **Management:** This category encompasses elements that are particularly relevant for management. It includes both tasks that require direct attention and decisions

from leadership, as well as key considerations or reminders of important practices. These elements are crucial for guiding MBSE adoption effectively, as management plays a central role in fostering and enforcing the necessary changes. It is also the largest grouping, reflecting the central role of leadership in driving organizational change.

- **Methodology, Language and Tools:** This category focuses on elements related to the core pillars of MBSE: modeling methodology, modeling languages, and modeling tools, as well as their integration and the provision of the necessary IT infrastructure. These components are essential for enabling the practical application of MBSE.
- **Modeling:** This category includes elements directly tied to the modeling process itself. As modeling is a cornerstone of MBSE, this group addresses a substantial number of challenges, pitfalls, and best practices that can directly impact the transition's success.

By organizing the findings this way, each level equips stakeholders with insights to navigate the transition with greater clarity and confidence, enhancing their capacity to build a robust and sustainable MBSE implementation. This chapter also serves as a reference complementing the maturity assessment tool, allowing readers to explore any elements not fully detailed in the visual assessment. To streamline usability in the assessment, references are omitted, and content is simplified to reduce visual load. Here, however, a comprehensive list of sources is provided for each element, facilitating deeper insight where needed and supporting users in exploring foundational studies related to the challenges, pitfalls, and best practices of each stage.

To provide an overview of the maturity assessment, we have included simplified versions of each level within the main text. These visuals illustrate the overall structure of the assessment, but the detailed content has been omitted due to the large size of the images, which cannot fit on a single page while remaining legible. The included images display only the headers of each element, with their detailed content described in the corresponding sections for each level.

For completeness, full versions of the maturity assessment visuals, including the instructions given and the self-assessment questions, are provided in the appendix. However, we acknowledge that these images may not be readable in printed form due to their size and level of detail. For readers of the digital version, we have included a link below to a Google Drive folder containing high-resolution images of each level. For optimal viewing, we recommend downloading the images, which allows for detailed examination.

Link: `https://drive.google.com/drive/folders/18JXsTCrmOKNdVTKgEfEe4foZF1OvXRdJ?usp=sharing`

5.1 Level 1: Initial Preparation

This level marks the foundational phase of any organization's transition to MBSE. It focuses primarily on preparing the workforce and management for upcoming changes, establishing a shared understanding and addressing early challenges. This stage sets essential prerequisites for a successful transition.

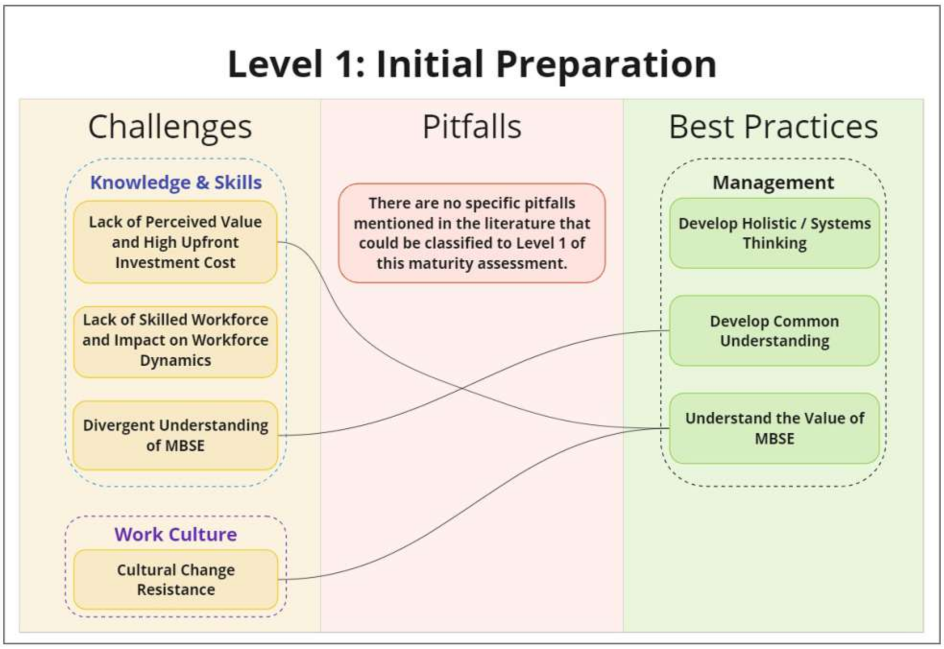


Figure 5.1: Simplified Overview of Maturity Assessment Level 1: Initial Preparation

Note: A complete version of this image, including the full text of each element, is provided in the appendix. However, due to the large size of the images and the limitations of standard page formats, the text may appear too small to read clearly in printed or standard PDF formats. For high-quality images that allow zooming and ensure readability, please visit the following link: `https://drive.google.com/drive/folders/18JXsTCrmOKNdVTKgEfEe4foZF1OvXRdJ?usp=sharing`

5.1.1 Challenges for Level 1: Initial Preparation

Knowledge and Skills

- **Challenge: Lack of Perceived Value and High Upfront Investment Cost**

  One of the primary challenges in adopting MBSE is the lack of perceived value among stakeholders, particularly in the early stages of implementation. The significant upfront investment required for MBSE, encompassing the establishment of models, tools, and trained personnel, can make it difficult to justify pilot projects or broader adoption initiatives. Stakeholders, especially those accustomed to traditional systems engineering methods, may be skeptical of the new approach, questioning whether it can deliver the same results as their familiar processes. This skepticism is often fueled by concerns over the unknowns associated with MBSE, such as its ability to produce the expected products and insights more efficiently or with higher quality. Generic claims of improved efficiency and effectiveness are typically insufficient to convince these stakeholders [49][21][41][7][6][34][73][74][57][75][42][76].

- **Challenge: Divergent Understanding of MBSE**

  Another key challenge in adopting MBSE is the diverse and often conflicting mental images and understandings of what MBSE truly means. Design engineers across various disciplines need a foundational grasp of the MBSE approach to harness its benefits effectively. However, there is often confusion surrounding the many related terms and concepts, such as Model-Based Engineering (MBE), Model-Driven Engineering (MDE), or Model-Driven Development (MDD), which can lead to fragmented understanding. Additionally, when some people hear "diagrams" or "models", they might think of static MS Visio diagrams or other forms of static representations that offer no functionality beyond a snapshot of the system, compared to the dynamic and integrative capabilities actual MBSE models can offer. Some engineering domains may struggle with the specific tools and languages used in MBSE, such as mechanical engineers who may find SysML behavior diagrams challenging to interpret, as they are more accustomed to CAD tools. This disparity in understanding can be worsened by established discipline-specific thinking and the widespread use of specialized, often homegrown models that do not align with the holistic systems approach MBSE promotes. For MBSE to be successfully implemented, it is crucial to ensure that all stakeholders are on the same page regarding its concepts, evolution, and intended benefits, fostering a unified approach to systems thinking and engineering [39][6][36][74][42][77][76].

- **Challenge: Lack of Skilled Workforce and Impact on Workforce Dynamics**

  Another challenge in the transition to MBSE is the general lack of skilled MBSE engineers within many organizations. Most companies do not have a readily available workforce that is equipped to easily transition to MBSE, as it requires a specialized set of skills and knowledge that many current employees may not possess [39][49][7][6][78][36][76].

  This skill gap can lead organizations to seek new talent, which can inadvertently threaten the job security and satisfaction of established workers who feel their expertise in traditional processes is becoming obsolete. Senior engineers, who are typically the technical experts in the current processes, may feel that the new tools and methodologies associated with MBSE undermine their existing knowledge base. This perceived threat can result in uncertainty and decreased job satisfaction, as the senior engineers struggle to see their place in the new MBSE workflow. Instead of demonstrating organizational commitment by leading the transition, these engineers might experience stress and show resistance due to the strain and uncertainty associated with the change. Additionally, varying levels of MBSE knowledge and experience across the workforce, often influenced by factors like age or individual adaptability, can lead to further resistance to change, compounding the difficulty of a smooth transition to MBSE [79][74].

Work Culture

- **Challenge: Cultural Change Resistance**

  Implementing MBSE often encounters significant cultural resistance within organizations. Engineers and stakeholders are traditionally accustomed to working with and reviewing documents, making the transition to models as primary artifacts a substantial shift in established workflows. Phrases like *"This is how we have always done it"* embody the inherent resistance to adopting new methodologies. This resistance is further compounded when key players lack adequate training and time to adapt, leading to a deficiency in understanding MBSE's value and processes. Such gaps not only foster skepticism but can also create a belief among employees that their systems are too large or complex to be effectively integrated into an MBSE framework. Without addressing these cultural and educational barriers, the transition to MBSE can face obstacles that hinder its successful adoption and integration. Recognizing and proactively managing this cultural resistance is paramount, as it stands among the greatest challenges organizations must overcome in their MBSE journey [39][80][5][49][36][73][75][42][76].

5.1.2 Pitfalls for Level 1: Initial Preparation

There are no specific pitfalls mentioned in the literature that could be classified to Level 1 of this maturity assessment.

We attribute this absence to the nature of the term itself. Pitfalls are defined as avoidable mistakes or traps that teams may fall into during the process of MBSE adoption. However, at this early stage, the focus is on establishing a common ground and building the right understanding, making it more about addressing fundamental challenges rather than avoiding missteps. These foundational aspects, like ensuring a mutual understanding of terms, goals and reasons, are so critical, they tend to be classified as challenges or best practices, rather than pitfalls.

5.1.3 Best Practices for Level 1: Initial Preparation

Management

- **Best Practice: Develop Holistic / Systems Thinking**

  An important best practice that has to be approached in the early phase is the development of a holistic / systems thinking mindset in the workforce. MBSE is grounded in systems thinking, which requires professionals from diverse disciplines, like software, mechanical or electrical engineering, to see the bigger picture of system development. While not everyone needs to become an expert or has to be concerned with the intricate details of specific domains, all engineers should understand and engage with system models to ensure collaborative progress. Changing the mindset of how the people think about developing a system. This foundational understanding of systems engineering principles is essential, especially for those new to the discipline. Without this shift in mindset and competency, the adoption of MBSE will struggle to succeed [80][81][7][6][82][83][76].

- **Best Practice: Develop Common Understanding**

  A best practice in the initial stages of MBSE adoption is to ensure a basic awareness and common understanding of MBSE concepts across the organization. Again, not everybody needs to become an expert in modeling and the concepts involved, but it is essential that all employees, especially those involved in the definition and deployment phases, receive foundational training to align on MBSE principles and terminology. This helps avoid misunderstandings and fosters a unified approach. Further training should be tailored, with deeper knowledge reserved for those directly involved in model development, while others focus on understanding and interacting with models. Without this fundamental common understanding, the organization risks creating silos, where only a subset of engineers embrace MBSE, leading to division and undermining the overall effort. Evaluation and follow-up training can help address knowledge gaps that may arise over time [73][81][7][6][82][83][84][85].

- **Best Practice: Understand the Value of MBSE**

  A best practice that is often mentioned, is making the advantages of the MBSE approach clear to everyone involved. It's essential that employees not only understand the theoretical benefits of MBSE, but also recognize how it can improve their own work, such as enhancing efficiency, quality, and reducing effort. This understanding helps build a unified workforce that supports the transition. Engineers are more likely to embrace the challenges of MBSE adoption if they can see its direct value in their daily activities. No amount of investment in tools and training will drive adoption unless the workforce perceives tangible benefits from using the MBSE approach. Targeted presentations and demonstrations of successful project experiences can further reinforce the value proposition and motivate engineers to collaborate towards successful adoption. This is crucial for fostering commitment and overcoming the inertia often encountered with organizational change [81][18][79][7][78][6][74][86][75].

5.2 Level 2: Planning & Structure

Once initial preparation is complete, this stage emphasizes strategic planning and structural decisions. Key actions include defining clear goals and scope, implementing progress metrics, managing expectations and establishing the necessary infrastructure and teams. This sets a stable foundation for subsequent modeling efforts and enables the launch of MBSE in pilot projects.

Figure 5.2: Simplified Overview of Maturity Assessment Level 2: Planning & Structure

Note: A complete version of this image, including the full text of each element, is provided in the appendix. However, due to the large size of the images and the limitations of standard page formats, the text may appear too small to read clearly in printed or standard PDF formats. For high-quality images that allow zooming and ensure readability, please visit the following link: `https://drive.google.com/drive/folders/18JXsTCrmOKNdVTKgEfEe4foZF1OvXRdJ?usp=sharing`

5.2.1 Challenges for Level 2: Planning & Structure

Management

- **Challenge: Lack of Leadership Buy-In**

  A critical challenge in adopting MBSE is the lack of leadership buy-in and support. Strong leadership and management commitment are essential to overcome the institutional inertia that often emerges when substantial changes like MBSE happen. Without active engagement, support, and sponsorship from leadership, any MBSE initiative is unlikely to gain the momentum needed for success. Leaders are key decision-makers in planning the way forward. Their involvement is crucial for setting priorities, allocating resources, and fostering an organizational culture that embraces MBSE. Insufficient preparation, persuasion, and commitment from management can severely undermine the introduction and integration of MBSE methods and tools, leading to a faltering or failed implementation [49][21][41][7][40][78][5][36][42].

- **Challenge: Measuring Progress and ROI**

  Another challenge in the adoption of MBSE is the difficulty in measuring progress and determining the return on investment (ROI). MBSE requires substantial upfront investment in tools, training, and processes, yet the benefits often take time to materialize, making it challenging to quantify immediate returns. To justify the investment and ensure continued support from stakeholders, it is crucial to establish clear metrics that track the progress of MBSE implementation within the organization. Additionally, a plan for measuring ROI should be developed, allowing for an assessment of how much value the company has gained from the transition over time. These measures are not only beneficial for evaluating the success of the adoption but also for making necessary adjustments to the strategy as needed, ensuring that the organization stays on track towards achieving its MBSE goals [6][78][5][75].

- **Challenge: Defining a Clear Scope and Goals**

  A critical challenge in adopting MBSE is the need to establish a clear and well-defined scope and set of goals for the effort. This clarity helps management as well as employees further understand the purpose and future value that MBSE should bring to their work. Without a well-defined scope, there is a risk of over-enthusiasm, where organizations may attempt to implement too many aspects of MBSE at once, leading to overwhelm and potential failure in the transition. Clear, achievable goals help manage expectations and ensure that MBSE is seen as a tool to enhance SE, not replace it. Executive leadership plays a crucial role in this process, as they must make informed decisions that shape the path forward, balancing short-term, low-cost adoption strategies with long-term, high-quality integration of MBSE. Additionally, managing user expectations is vital, as sales personnel of tools and trainings often make ambitious promises that can lead to unrealistic hopes. A common understanding of the *"why"* and *"what"* of

MBSE adoption, supported by strong leadership, is essential to ensure a successful transition [80][5][78][5][36][75][76].

Knowledge and Skills

- **Challenge: Steep Learning Curve**

  The adoption of MBSE is often hindered by a steep learning curve, especially for employees with little to no prior experience in modeling. Training is essential to ensure that users can effectively utilize MBSE tools and methodologies, but this training demands significant time and resources from both employees and management. In some cases, training has been rushed or even skipped, under the false assumption that MBSE methods are self-explanatory, leading to further complications down the line. Engineers without a background in modeling face particularly frustrating challenges, as they must quickly grasp complex concepts and techniques. The costs associated with proper training, including both the financial investment and the time commitment, are necessary burdens that organizations must bear to build a basis for a successful transition towards MBSE. Without adequate training and support, the steep learning curve can impede progress and hinder the overall adoption of MBSE practices [49][87][6][78][34][57][84][75].

Methodology, Language and Tools

- **Challenge: Choosing and Integrating Modeling Methodology and Tools**

  Selecting the right modeling methodology and tools is a difficult challenge in the transition to MBSE. The methodology serves as the guide for implementation, defining rules, guidelines, and processes that will be followed. Setting it up, documenting it, and creating supporting materials like training guides can be a resource intensive task. Customizing the method to fit the organization's specific scope and purpose can further complicate this task. Additional extensions or adaptations to the methodology down the line can be challenging to implement [5][84][75].

  Equally important is the careful selection of the modeling tool. No single tool can satisfy all needs across the entire product life cycle, so it's crucial to choose a tool that can integrate well with the company's existing tool landscape. This integration is vital because different tasks will still require specialized tools, necessitating harmonized interfaces for effective data exchange. Tool incompatibility, lack of interoperability, and communication issues between different tools can significantly hinder MBSE adoption [5][81][6][34][36][88][84][75] Nevertheless, focusing too heavily on tools rather than on the methodology can lead to misaligned efforts [36].

5.2.2 Pitfalls for Level 2: Planning & Structure

Management

- **Pitfall: Fast False Start**

  A common pitfall in MBSE adoption is starting too quickly with too many initiatives at once, leading to overwhelm and potential failure. This "fast false start" can cause skepticism within the organization, reinforcing the belief that modeling doesn't work. To avoid this, it's important to introduce MBSE gradually, in smaller, manageable steps. This allows for a smoother transition, giving time to address any issues that arise and ensuring a more successful long-term adoption. Think big; start small [89][15].

- **Pitfall: Underestimating the Learning Curve and Expert Involvement**

  Another pitfall in MBSE adoption is underestimating the steep learning curve and over-relying on external experts. While experts can provide valuable guidance, the organization must take ownership of the process to ensure lasting success. Experts should assist during the transition but only be consulted selectively, allowing the organization to build internal expertise. Expecting immediate efficiency gains from MBSE, especially under project pressure, is unrealistic. Proper time and planning are essential to allow for learning and gradual improvement [89].

- **Pitfall: Missing IT Infrastructure**

  Failing to provide the necessary IT infrastructure to support the modeling environment can be a pitfall. Ensuring that the network, processing power, and storage capacity meet the demands of the modeling tools is essential. Without sufficient IT infrastructure, performance issues can arise, leading to delays, inefficiencies, and frustration among users, ultimately hindering the MBSE adoption process [87].

5.2.3 Best Practices for Level 2: Planning & Structure

Management

- **Best Practice: Focus on the purpose of the MBSE adoption**

  A successful transition to MBSE depends on maintaining a clear focus on the purpose behind the change. Defining and communicating the goals, scope, and expected benefits of MBSE adoption from the outset is crucial. A clear end goal helps guide decisions throughout the process and prevents the effort from being sidetracked by new ideas that do not align with the original needs. While new concepts may arise during implementation, these should be carefully evaluated to ensure they do not divert attention from the core purpose. Organizations need to spend as much time understanding the problem space as they do developing solutions. By clearly articulating the purpose, the perceived complexity of MBSE can be minimized. The purpose should address key questions that the model is designed to answer and ensure that every stakeholder understands how MBSE aligns with business objectives. Communicating the "why" helps manage expectations and ensures that everyone involved remains focused on solving the right problems, preventing costly misdirection in the effort [80][73][81][21][41][18][7][78][6][82][86][83][85][42].

- **Best Practice: Leadership and Management Buy-In**

  The success of MBSE adoption depends on strong leadership at multiple levels. The systems engineer tasked with leading the MBSE transition must play a leadership role, setting clear objectives, guiding the team, and managing the pace of change to ensure it is embraced by the workforce. Leadership is about understanding both people and processes, knowing when to push forward and when to hold back enthusiastic ideas until the organization is mature enough to continue with the next step. At the same time, buy-in from upper management is critical to provide the necessary resources, authority, and alignment of business goals with the MBSE transition. Management support is a key enabling factor, as it ensures a unified commitment across the organization, sustains momentum, and addresses concerns such as ROI and resource allocation, making MBSE adoption both possible and sustainable. Communication is key in this context, as convincing leadership can be challenging. It's essential to focus on *why* MBSE benefits the business rather than the *how* of its technical implementation [39][80][79][86][7][78][6][75][42].

- **Best Practice: "Think Big, Start Small, Evolve"** [80]

  A strategic MBSE approach should begin by identifying clear, long-term goals, but manage the uncertainty and complexity of the transition by introducing small, manageable steps. Starting with a small, highly motivated group or a pilot project allows the organization to experiment in a controlled environment, learning from the specific challenges of the involved processes, domains, and tools. This approach helps to identify and resolve issues early, creating a foundation for broader adoption. Pilot projects serve as valuable trials where limited but targeted applications of MBSE can demonstrate tangible benefits. For instance, modeling just a few structural or behavioral elements needed for a specific capability can provide immediate value, which encourages expanding the modeling efforts incrementally. Each success builds confidence, skills, and management buy-in, helping the organization to progressively move from small-scale implementations toward an enterprise-wide MBSE approach. This incremental approach is key; crawl, walk, then run. Focus on achieving early, frequent successes to build momentum and confidence, ensuring steady progress and long-term sustainability [80][81][18][90][7][78][6][84][85][42][91].

- **Best Practice: Change Process Management**

  Successfully transitioning to MBSE requires more than just technical expertise, it demands experienced guidance in managing the change process itself. Engaging professionals who have navigated similar transitions significantly increases the chances of success. These experts help continuously assess progress, identify training needs, and adapt strategies based on lessons learned. Effective change management includes fostering a culture of openness, evaluating and documenting business processes as well as providing ongoing coaching beyond initial training. This ensures that the organizational shift towards MBSE is well-supported, gradual, and adaptable to challenges [80][81][79][7][6][82][86][42].

- **Best Practice: Establish a MBSE Core Team**

  Creating a dedicated MBSE team with clearly defined roles is frequently highlighted as a best practice for successful adoption. This core group should include designers with tailored training in the methodology and modeling language, model users trained to interpret system models, and tool owners proficient in MBSE tools who are responsible for adapting and evolving the tools to meet organizational needs. Additional key roles often mentioned in the literature include MBSE specialists, commonly referred to as MBSE Champions, who possess a strong modeling background and a deep understanding of their business unit's domain. These champions foster a network of experienced peers, guide and support the adoption process, and drive cultural change by attracting followers, securing resources, and ensuring corporate commitment. Champions often act as advisors to engineers starting their own projects, particularly when other support structures are limited. Another critical role is the model curator, tasked with optimizing the MBSE environment to enable accurate system analysis. In a single-source-of-truth solution, effective model curation is the foundation for reliable system decisions, as most, if not all, information derives from this source. Enterprises adopting MBSE that fail to address this personnel related practice, risk undermining their ability to execute their vision. Required skills and experience might not be readily available, but understanding and planning for changes in organizational structure, responsibilities and competencies is critical. By acquiring new specialists or providing the right level of training for each role, the complexity of MBSE can be significantly reduced for all stakeholders. With ongoing support and a clear division of responsibilities, the core MBSE team becomes a crucial driver of organizational alignment, ensuring that MBSE practices are effectively integrated and continuously improved across projects [73][18][46][7][72][78][6][57][86][84][75][85][42].

- **Best Practice: Knowledge Transfer and Collaborative Learning**

  Facilitating knowledge transfer through mentoring, online forums, and peer social networks can be used to empower engineers to learn and adopt MBSE together. This collaborative approach helps bridge the gap between experienced engineers, who are less likely to be willing to embrace a change to established norms, and younger engineers more familiar and eager with digital tools [74]. Encouraging social interaction among learners fosters a supportive learning environment, increasing motivation and improving outcomes. By integrating both formal and informal knowledge sharing mechanisms, organizations can overcome resistance, enhance digital literacy, and sustain MBSE learning throughout the team [79][78][74][92].

- **Best Practice: Define Measures of Progress**

  Establishing effective metrics to track the progress of MBSE adoption is particularly important given the delayed return on investment. Typical lower-level systems engineering metrics may not be as meaningful in an MBSE environment. For example, MBSE might reveal more requirement allocation errors, but this could be due to better error discovery rather than an increase in errors themselves [21]. Therefore, traditional measures might not fully capture the impact of MBSE. A dedicated MBSE business case should be coupled with business criteria, focusing on progress indicators such as engagement with MBSE tools and quality improvements. Regularly tracking these tailored metrics allows for immediate feedback and adjustment of strategies, if necessary [73][21][41][7][78][6][55][54][86]. A specifically valuable resource in this context might be [54], who not only address the importance of measuring progress, but provide in-depth guidance on which measurements could be helpful to measure benefits and ultimately the success of the MBSE approach.

- **Best Practice: Accept the Learning Curve**

  Organizations starting their MBSE transition have to accept and embrace the learning curve. When deciding on the pilot projects, it is important to choose initiatives that can accommodate the necessary learning efforts without jeopardizing critical timelines or business objectives. Otherwise employees might drop MBSE techniques in favor of established development methods due to the time pressure. Projects should be budgeted not only for delivering artifacts but also for covering the costs associated with this learning phase. Stakeholders additionally have to accept some initial redundancies as models are developed to capture data that is already contained in other sources (e.g. documents, presentations, etc.). The transition requires time and patience. Adequate training and supportive work relationships are vital to reducing uncertainty and resistance, thereby fostering job satisfaction and commitment to the change. Engineers need structured training on new digital engineering tools and modeling techniques to build confidence and expertise. By allocating dedicated time for learning and providing access to experts for guidance, organizations can mitigate stress and fatigue, preventing disengagement. Ultimately, accepting and supporting the learning curve helps create a positive environment where employees can adapt and thrive during the transition to MBSE, leading to a more successful and sustainable implementation [81][18][79][78][6].

- **Best Practice: General Planning and Management Proposals**

  Organizations must approach the transition towards MBSE with the same rigor as any new technology, ensuring that the necessary knowledge and quality assurance are in place. A structured governance framework helps monitoring progress, promoting transparency, and maintaining accountability. Clear modeling guidelines, encapsulating the MBSE language, methods, tools, and personnel roles, should be documented in accessible formats, such as a model management plan or a collaborative wiki. Additionally, defining the scope and responsibilities of systems engineering within the organization fosters clarity and alignment across teams. Another important aspect is to establish coordination mechanisms at both technical and organizational levels to facilitate communication regarding domain, tools, and digital strategies. Organizations should remain committed to their vision, be resilient, learn from both successes and setbacks and continuously adjust the course to deliver incremental, demonstrable results. A culture of open communication and support can help guide their teams through the complexity of MBSE adoption [73][78][93][86][84][42].

Methodology, Language and Tools

- **Best Practice: Capable IT Infrastructure**

  A robust and scalable IT infrastructure is fundamental to supporting MBSE adoption. The infrastructure should provide sufficient computing power, network performance, storage capacity, and access controls to handle the complexities of model data throughout the system's life cycle. Failure to ensure an adequate IT setup can result in slowdowns, system bottlenecks, or even interruptions in model management. Proper infrastructure ensures that modeling tools can operate efficiently, preventing delays in deliveries and enabling smoother collaboration across the organization [39][87][78][6].

- **Best Practice: Tool Access**

  Providing widespread access to MBSE tools is important. Failure to do so can create divisions within the engineering team, jeopardizing the adoption of MBSE. Not every engineer will need the same level of access, but ensuring that those who need the tools, whether for model production or consumption, have appropriate licenses and access is essential. The cost of tool acquisition and licensing can be significant, so careful planning must account for both the needs of the organization and cost-effective solutions. Thoughtful distribution of licenses, aligned with roles, can lower costs and enhance collaboration [81][6][57][84].

- **Best Practice: Tool Selection and Integration**

  Ideally, an MBSE adoption effort considers the methodology and modeling language before selecting and integrating modeling tools. Many organizations make the mistake of focusing on tool acquisition too early, which can limit the benefits of MBSE if the tools do not fully align with the project's needs. First, the capabilities the organization would like to develop by using MBSE should be identified, in order to decide on the tools that will help it fulfill its goals. Then, proper tool integration with the existing tool landscape is crucial to ensure communication and data exchange between different tools across various engineering domains. Rushing into tool selection without understanding the organization's modeling needs can lead to inefficiencies, limiting the potential impact of MBSE [81][21][7][89][6][82].

5.3 Level 3: Pilot Projects

At this stage, organizations begin applying MBSE to specific pilot projects. This is the first level to involve active modeling, requiring strategic decisions and preparatory steps to avoid early missteps. Key objectives include maximizing the return on pilot efforts, ensuring best practices in modeling and building a base for sustainable MBSE use across future projects.

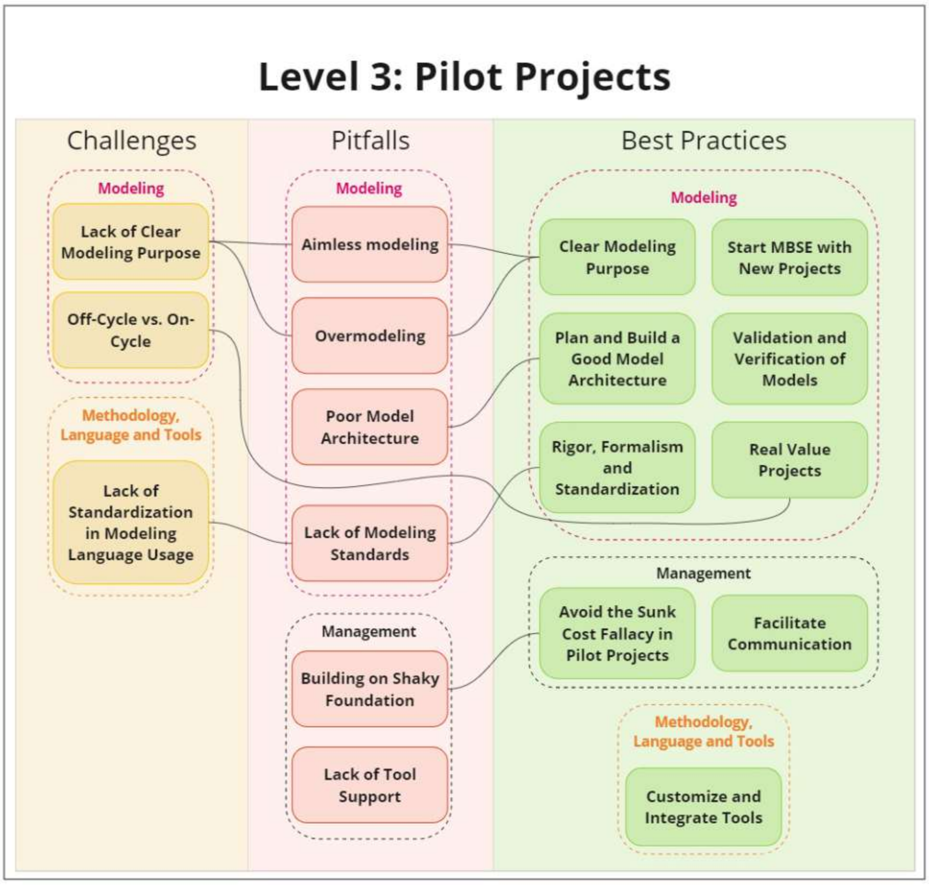


Figure 5.3: Simplified Overview of Maturity Assessment Level 3: Pilot Projects

Note: A complete version of this image, including the full text of each element, is provided in the appendix. However, due to the large size of the images and the limitations of standard page formats, the text may appear too small to read clearly in printed or standard PDF formats. For high-quality images that allow zooming and ensure readability, please visit the following link: `https://drive.google.com/drive/folders/18JXsTCrmOKNdVTKgEfEe4foZF1OvXRdJ?usp=sharing`

5.3.1 Challenges for Level 3: Pilot Projects

Modeling

- **Challenge: Lack of Clear Modeling Purpose**

  As organizations progress to the stage of pilot projects and begin actual modeling efforts, one key challenge is defining the specific purposes of each model. While high-level MBSE goals are clear, many struggle to ensure that individual models are tailored to answer well-defined questions or meet specific project needs. Without this clarity, there is a risk of overmodeling or creating models for the sake of modeling, which can lead to inefficiencies and models that are not as effective or communicative as they could be [21].

  Additionally, models should be designed with the users in mind. Many people across various disciplines will interact with the models, and not all have the same level of knowledge or expertise. Querying the model often requires additional training. Overcomplicating models or cramming too much into one can further impede navigation and understanding, leading to overlap with other models and resulting in inconsistency and inefficiency. Therefore, it's important to keep models simple and purposeful, ensuring they are easy to navigate and able to accomplish their primary objectives [5][41].

- **Challenge: Off-Cycle vs. On-Cycle**

  Deciding on the beginning modeling effort whether to introduce MBSE during an actual project with real clients and deliverables (on-cycle) or to test it in a controlled, off-cycle sandbox environment can be challenging. In an on-cycle adoption, the pressure is high, as there are real-world consequences, timelines, and client expectations. This can lead to rushed implementation and difficulty addressing unexpected challenges. Conversely, off-cycle adoption provides the team with space to experiment and refine their MBSE approach without the pressure of a live project. However, the gap between sandbox experimentation and applying MBSE in real-world projects can create difficulties in translating theoretical knowledge into practical execution [5][6].

  This transition problem is often described by the "blank page syndrome", where individuals struggle to begin creating meaningful models for real systems after learning MBSE theory and practicing on generic examples. Unlike working on predefined examples, applying MBSE to the specific needs of the target system can feel overwhelming [34][75].

Methodology, Language and Tools

- **Challenge: Lack of Standardization in Modeling Language Usage**

  SysML is often seen as the de facto standard for MBSE, but its complexity and vast range of possibilities can be overwhelming, especially for new users. Regardless of what modeling language is chosen, but especially when it is similar in complexity to SysML, it is important to realize that not all features need to be utilized, it is crucial for organizations to establish a company-wide standard for how the modeling language will be used. By restricting the use of certain features and keeping models simple, organizations can make MBSE more accessible to stakeholders and ensure that models remain clear and functional. Without those standards, the risk increases that models will vary significantly in style and structure, making it challenging for colleagues or stakeholders to understand and work with them effectively [80][36].

5.3.2 Pitfalls for Level 3: Pilot Projects

Modeling

- **Pitfall: Aimless modeling**

  A key pitfall when modeling is the failure to define a clear purpose and scope for each model. Without a specific goal, modeling can become aimless, leading to overcomplicated models that do not effectively answer stakeholder questions. The focus should be on system engineering rather than modeling. A model should be driven by its ability to address engineering questions and needs. If a model does not satisfy the requirements of stakeholders or provide the necessary engineering insights, it is incomplete. This lack of focus can lead to delivering models that seem finished, but fail to provide the required engineering answers. Effective modeling requires a clear understanding of competency questions and the scope of the model to ensure that it serves its intended purpose [89][6][94][86][15].

- **Pitfall: Lack of Modeling Standards**

  Another pitfall is the absence of modeling standards. Without standardized rules for modeling practices, there is a risk of developing highly sophisticated and unique models that are not easily understood or integrated across the organization. Often, a small group of modelers may create detailed models using complex or non-standard elements, leading to "Ivory Tower" scenarios where these models are comprehensible only to their creators. Such models, while technically accurate, can become isolated from broader development processes, diminishing their utility and impact. This lack of standardization makes collaboration and integration challenging. Implementing clear, company-wide modeling standards is crucial to ensure models are understandable, consistent, and valuable across the organization [87][89].

- **Pitfall: Overmodeling**

  Overmodeling is a typical pitfall during the beginning modeling effort of MBSE. The extensive capabilities of the modeling languages (e.g. SysML) lead to excessively detailed models. This can result in models that are cumbersome and difficult to manage, as each additional model element requires significant follow-up work, such as updates to definitions, diagrams, and simulations. Overmodeling not only increases the complexity of the model but also diminishes its usefulness, as the effort to maintain and update the model often outweighs its benefits. Effective modeling should prioritize critical system components and avoid mixing different engineering levels within the same model, as this can lead to significant refactoring and inefficiencies [89][6][86][15].

- **Pitfall: Poor Model Architecture**

  Another critical pitfall in MBSE adoption is poor model architecture, which can manifest as large, monolithic models with excessive usage levels and circular references between projects. Sometimes multiple engineering levels are combined within a single model under the illusion of simplicity. For example, merging the architectural design of a system with that of its subsystems within the same model. This can lead to significant complications, such as cumbersome refactoring and integration difficulties. Effective model architecture requires clear boundaries and a well-structured organization to maintain control and comprehensibility throughout the development process. Without these structures, issues such as duplicate elements, broken traceability and others can arise. Ensuring a well-organized model architecture with defined package structures and clear separation of concerns is essential to avoid this pitfall and support a successful MBSE implementation [87][6][94][95][15].

Overmodeling and Poor Model Architecture are closely related, but they address different aspects of MBSE and are therefore kept separate. Overmodeling focuses on the content of the models, warning against adding too much detail or including unnecessary elements, which results in complexity and maintenance challenges. It's about knowing when to stop and what level of detail is necessary. Poor Model Architecture, on the other hand, is about the structure of the models, emphasizing the need for a well-planned and organized approach to the model's architecture. It highlights issues like combining different engineering levels or failing to separate concerns properly, which leads to inefficiencies in managing and navigating the model. They complement each other but deal with distinct problems: one is about how much to model, and the other is about how to organize the model.

Management

- **Pitfall: Building on Shaky Foundation**

  Relying on the models from pilot projects as a foundation for further development can be a pitfall. There is a high risk in building on a shaky foundation to avoid the cost of establishing a stronger, more reliable base. It's crucial to recognize that pilot models are often preliminary and may require significant refinement. Stakeholders should be cautious about the temptation to extend these initial models without addressing underlying issues, as this can perpetuate foundational weaknesses and impact the overall effectiveness of the MBSE approach [21].

- **Pitfall: Lack of Tool Support**

  Starting modeling efforts without a local tool support team can be a significant pitfall. Engineers need a reliable environment to work effectively. Otherwise, issues with tools and infrastructure can hinder progress and lead to frustration. Without proper setup and support, users may struggle with tooling difficulties, which can derail their objectives and result in misplaced blame on the MBSE adoption effort, rather than addressing the root causes of the problems [86].

5.3.3 Best Practices for Level 3: Pilot Projects

Modeling

- **Best Practice: Clear Modeling Purpose**

  When embarking on any modeling initiative, it is essential to clearly define the model's purpose. These purposes should be articulated as specific questions that the model is intended to answer. This way the model is clearly finished when all relevant questions can be answered, which ensures that models do not get overmodelled and unnecessarily complex, helping to prevent wasted effort, frustration, and low recognition of the model's benefits. If models are created without clear goals tied to supporting key engineering activities, they risk delivering little value and generating low stakeholder buy-in. By aligning each model with specific, measurable outcomes and reducing unnecessary maintenance efforts, organizations can ensure that MBSE drives meaningful results and supports business objectives effectively. Clear modeling objectives provide the necessary focus to produce knowledge, streamline engineering tasks, and deliver tangible benefits, rather than creating models for modeling's sake [80][73][21][41][86][83].

- **Best Practice: Plan and Build a Good Model Architecture**

  A best practice in early modeling efforts for the MBSE adoption is to carefully plan the model architecture to maintain clarity and manage complexity. Large, overly complex diagrams can hinder understanding and productivity, while the true strength of MBSE lies in the ability to query models and generate specific views / diagrams or reports as needed. Views / diagrams, just as the model itself, should be purpose-driven, focusing on answering specific questions with only the necessary information included. To manage complexity, diagrams should be structured hierarchically, breaking down large systems into smaller, more digestible diagrams, capturing key concepts at each level. However, it has to be acknowledged that model architecture and modeling are iterative processes. Early in the model development life cycle, decisions around both the scope (breadth) and level of detail (depth) should be explored simultaneously, ensuring the architecture can support future growth and prevent technical debt from building up. Researching and testing different levels of depth and breadth helps uncover architecture drivers, such as differences in system structure or direct interfaces between components deep within the hierarchy, and allows for continuous refinement based on feedback. This iterative approach ensures flexibility as the model grows, allowing the architecture to evolve through input from stakeholders and engineers. The goal is not to perfect the model or architecture from the outset, but to iteratively improve it based on real-world usage, gradually expanding its value and relevance for the entire organization. A well-planned architectural vision, especially using reference architectures, ensures a logical structure that aids communication, stakeholder engagement and scalability, ultimately making the model more effective as it evolves [21][41][93][42][18].

- **Best Practice: Rigor, Formalism and Standardization**

  Establishing formal guidelines and standardization in modeling is critical, especially in the early stages of MBSE adoption. Modeling languages such as SysML provide incredible flexibility, but without formalized guidelines, the resulting models can vary widely between modelers, which can become overwhelming for teams with limited model / modeling experience, reducing their effectiveness and leading to confusion. To ensure consistency and interoperability across teams, formal modeling guidelines, standardized rules, and simplified language (e.g. SysML) constructs should be adopted early. These constraints help streamline the modeling process, improve clarity, and make it easier for all stakeholders to understand and work with the models. As the number of users grows, the need for rigor increases, and having standardized practices in place ensures scalability and reduces complexity. Creating enterprise-wide modeling standards promotes coherence and alignment across projects, making the transition to MBSE smoother and more efficient [80][72][6][86][73][21][87][78][34][55][93][57].

- **Best Practice: Start MBSE with New Projects**

  Beginning MBSE adoption with a new project is often a more effective approach than converting existing document-based projects. New projects eliminate the risk of importing outdated workflows, errors, or shortcuts that can undermine the learning and application of MBSE. By starting from scratch, teams focus on learning the methodology without relying on existing artifacts that may compromise the MBSE process. Moreover, existing projects often have fixed budgets and deadlines that do not account for the additional time and resources needed for MBSE training, making it harder to succeed under those constraints [81][7]. However, not every company has a new project suitable for the beginning of this transition. In those cases, it can be a best practice to start with converting well-documented, reusable artifacts from previous projects. These detailed artifacts allow developers to focus on learning the MBSE methodology rather than on content creation. Furthermore the existing artifacts can be used for verification and validation of the new models. As long as the risks of this approach are understood and measures taken to avoid translating errors of past projects into MBSE, this approach can be used and adds most value when the artifacts are likely to be reused in future MBSE efforts, ensuring the effort contributes beyond just learning purposes [81].

- **Best Practice: Real Value Projects**

  Another often mentioned best practice for MBSE adoption is to start with real projects that bring tangible value to the organization, rather than mock-up or sandbox projects. Working on real projects with actual deadlines and deliverables enhances motivation, as engineers see their efforts contributing directly to the organization's goals. In a real-world setting, employees are required to fully engage with MBSE, ensuring they learn and apply it in a meaningful way. If the project is purely for training, there's less urgency, leaving room for incomplete learning and procrastination. Real projects, aligned with the organization's domain, help highlight MBSE's practical benefits and demonstrate its value, motivating stakeholders to commit to the transition. This is further amplified when the focus lies on high-impact use cases, for example change impact analysis. These tasks that are time-consuming and error-prone when done with traditional methods can shift people from skepticism, when the benefits of MBSE are demonstrated with them. Additionally, using MBSE to generate real deliverables such as documents, reports etc. further strengthens adoption by showing how MBSE can streamline daily activities and improve communication among project stakeholders. Providing real-world success examples can add to justifying the initial investment and help gain broader acceptance within the organization [81][21][41][78][6][75].

- **Best Practice: Validation and Verification of Models**

  Ensuring the accuracy and quality of models from the start is critical. To maintain model consistency and completeness, it is essential to implement both manual and automated validation processes. Manual validation can involve regular model reviews that include not just the model creators but also domain experts to ensure all relevant stakeholders are engaged. Automated validation tools can further help by applying rules to check for model consistency and completeness, reducing errors early in development. Properly validated models give stakeholders confidence in their predictive capabilities, supporting early virtual validation of system performance and behavior. Additionally, involving subsystem stakeholders in co-engineering activities ensures that transitions between system levels are accurate, feasible, and align with the perspectives of all engineering teams. This ongoing validation throughout the system lifecycle prevents issues later on and solidifies the model's role as a reliable reference across the organization [73][87][86].

Management

- **Best Practice: Facilitate Communication**

An essential best practice for management during MBSE adoption is ensuring effective communication across all stakeholders. A clear communication plan should specify who communicates what, how, and when, with follow-up protocols to ensure alignment and accountability. Regular updates on the transition's timeline, strategy, and objectives, shared through tools such as webinars, emails, and meetings, foster engagement and ensure the plan remains realistic. Open and frequent communication of risks, protocols, and progress across departments is vital, with digital artifacts serving as effective tools for sharing information. Close coordination between the MBSE development team and systems engineers is particularly critical early on to align expectations, manage scope, and address trade-offs between implementation feasibility and immediate system engineering needs. While MBSE provides a single source of truth to streamline communication, the full potential of this benefit can only be achieved when management actively promotes and facilitates these practices, ensuring transparency, collaboration, and shared understanding throughout the organization [80][79][7][78][6][84].

- **Best Practice: Avoid the Sunk Cost Fallacy in Pilot Projects**

  Another best practice in MBSE adoption is recognizing that pilot projects are primarily for learning and experimentation, not for creating final models. Organizations should approach pilot efforts with the mindset that the value lies in the lessons learned, the insights gained, and the experience of using system modeling tools, not in the models themselves. If the pilot toolset or models prove unsuitable for long-term goals, be willing to change direction or tools. By treating pilot models as prototypes and focusing on gaining expertise, organizations can avoid the fallacy of building on an unstable foundation simply to preserve prior investments [21][41]. Additionally, when using prototypes, scaling must be carefully considered. Prototypes that cover too small a scope may not reveal critical issues that arise at larger scales, leading to false assumptions about the feasibility of MBSE implementations. Prototyping is valuable, but ensuring the prototype reflects the appropriate scope is essential for identifying potential challenges [80]. Having this in mind allows for the flexibility to ensure that future MBSE efforts are built on validated, reliable solutions, rather than prematurely building on top of pilot outcomes.

Methodology, Language and Tools

- **Best Practice: Customize and Integrate Tools**

  Another important step in MBSE adoption is customizing and integrating the chosen modeling tool into the organization's existing toolchain. This process typically involves extending / limiting the tool's language capabilities, implementing validation rules, and developing automated scripts to align the tool with both the modeling methodology and organizational needs. Customization efforts could for example work on optimizing the tool by making frequently used elements easily accessible while removing unused ones. The main challenge often arises in integrating the MBSE tool with other essential engineering tools, e.g. those used for testing and simulation. Ensuring interoperability through open interfaces is vital, as it facilitates seamless import and export processes across different design phases and prevents reliance on proprietary tools, which can cause issues down the road. In this context, it can be very helpful to have technically knowledgeable support staff assisting engineers. These well-defined interfaces and consistent modeling standards across teams will help tremendously for successful integration. By integrating the MBSE tool effectively, organizations can simplify workflows, reduce the need for redundant tools, and improve communication and collaboration across teams [73][81][89][6].

5.4 Level 4: Scaling MBSE Adoption

After pilot projects have been completed, this final stage focuses on scaling MBSE adoption across the organization. Here, the focus is on expanding MBSE application, improving modeling efficiency and establishing robust maintenance and long-term management of models. This stage aims to secure the long-term value and success of the MBSE approach by fully integrating it within organizational processes.

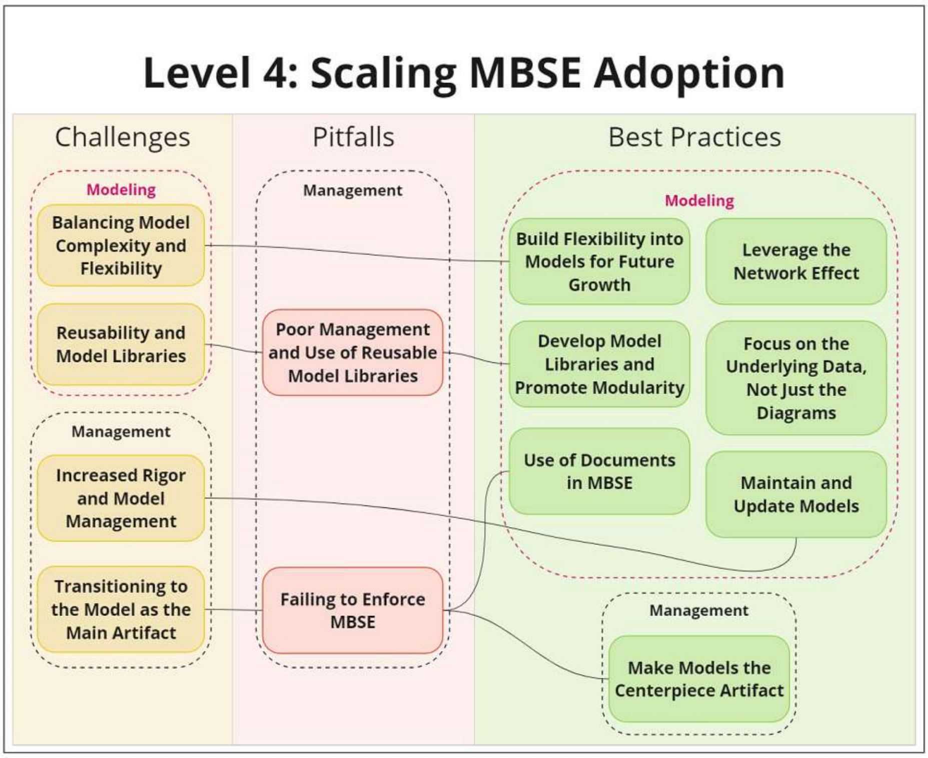


Figure 5.4: Simplified Overview of Maturity Assessment Level 4: Scaling MBSE Adoption

Note: A complete version of this image, including the full text of each element, is provided *in the appendix. However, due to the large size of the images and the limitations of standard page formats, the text may appear too small to read clearly in printed or standard PDF formats. For high-quality images that allow zooming and ensure readability, please visit the following link:* `https://drive.google.com/drive/folders/18JXsTCrmOKNdVTKgEfEe4foZF1OvXRdJ?usp=sharing`

5.4.1 Challenges for Level 4: Scaling MBSE Adoption

Management

- **Challenge: Transitioning to the Model as the Main Artifact**

  Shifting from documents to the system model as the primary artifact can be difficult for stakeholders not yet comfortable with the model-based approach. Decisions must be made on when to generate documents versus conducting reviews and meetings directly within the model using the modeling tool. Reviews conducted directly from the models can have limited success, especially with external experts who are not proficient with the modeling tools. Tailoring model views to specific domains is necessary to make the information digestible for various stakeholders. Until the automatic generation of gate products from models is fully mature, it is important to allocate adequate schedule time and margin for document generation, as technical issues may arise that need to be addressed before delivery [80][34][44][76].

- **Challenge: Increased Rigor and Model Management**

  Models are not static, they require continuous management, updates, and data integration to remain relevant and useful throughout the product life cycle. Without verification, validation, configuration control, and quality checks, the value of models can quickly diminish. To ensure models remain trusted engineering artifacts, organizations must commit to rigorous planning and processes, treating the MBSE environment as a key component of the digital engineering ecosystem [87].

Modeling

- **Challenge: Reusability and Model Libraries**

  When starting the modeling effort, there is no established library of reusable system models. Everything must be built from scratch. This process is time-consuming, especially for inexperienced modelers who may need training or rely on the limited availability of experienced modelers. Reusability isn't simply copying and pasting models from one context to another, as this approach compromises the "single source of truth" once either the source or target model is modified. Establishing modularity and reusability requires careful planning to ensure consistency and efficiency across the system models [5][74][75].

- **Challenge: Balancing Model Complexity and Flexibility**

The challenge lies in finding a balance between creating models that address immediate needs without overcomplicating them, while also ensuring they are flexible enough to accommodate future applications and growth. Overly simplistic models may require large revisions as project requirements expand, whereas overly complex models can become inefficient and difficult to maintain. Building models with an eye toward future adaptability helps manage technical debt and ensures their longevity within the MBSE ecosystem [21][41].

5.4.2 Pitfalls for Level 4: Scaling MBSE Adoption

Management

- **Pitfall: Failing to enforce MBSE**

  When MBSE is starting to become a central part of development another pitfall can emerge. When MBSE is encouraged but not enforced, it can happen that both approaches are used simultaneously, leading to inconsistent practices within a project. Some team members may embrace modeling, while others continue working with traditional document-based methods. This disjointed approach can cause confusion and inefficiency, particularly when deadlines approach. Under pressure, people often revert to familiar ways of working, abandoning the MBSE transition. This creates a "Model Island" where the models exist in isolation, disconnected from the rest of the development process. For MBSE to succeed, modeling must be fully integrated and mandatory, replacing traditional activities and ensuring that all stakeholders are aligned with the model-based approach. If left optional, modeling efforts are often the first to be discarded under pressure, undermining the transition to MBSE and reducing its potential benefits [89]

- **Pitfall: Poor Management and Use of Reusable Model Libraries**

  The pitfall of poor management and enforcement of reuse libraries occurs when libraries of reusable models are not properly maintained or consistently enforced. Without clear guidelines, teams may fail to use existing models effectively, leading to duplication of effort, inconsistencies, and wasted time. The effective management of reuse libraries ensures that models are accessible, standardized, and used across projects [87].

5.4.3 Best Practices for Level 4: Scaling MBSE Adoption

Modeling

- **Best Practice: Leverage the Network Effect**

  An important best practice at this stage is to capitalize on the network effect by making models relevant to a wide range of stakeholders. As more stakeholders engage with the model, its value increases through broader adoption and reuse as well as a broader audience of users fueling its continuing maintenance and evolution. Integrating and using models across collaborating systems allows for better insights into interdependencies at the enterprise level, which can improve decision-making. By expanding the scope of the model to address the needs of various stakeholders and system life cycle phases as well as making it a recognized reference, organizations can drive greater involvement and amplify the benefits of MBSE. This network effect accelerates the adoption of MBSE, as the initial investment in model creation is offset by the model's growing utility and value across the organization. The vision is to create a system representation that allows various stakeholders to extract artifacts for their own specific use [21][81][41][86][75][39].

- **Best Practice: Focus on the Underlying Data, Not Just the Diagrams**

  While the diagrams are often the most visible and tangible products of the MBSE approach, the real power and value lie in the underlying data that supports those diagrams. The true potential of MBSE comes from how it captures, integrates, and manages a multi-layered network of elements, attributes, and relationships in electronic format. This data allows the system to be queried, analyzed, and manipulated in ways that traditional methods cannot achieve, enabling the creation of customized views that trace relationships between components, requirements, or functions. It is crucial to build models with a focus on supporting data input, management and exchange, ensuring that the data is easily accessible and usable. The more data a model integrates, the more valuable it becomes for representing, simulating, and validating different aspects of the system. This ensures that models are not just static visual representations but dynamic tools that enable deeper insights and decision-making across the system's lifecycle [21][41][6].

- **Best Practice: Maintain and Update Models**

  Once MBSE has been adopted beyond pilot projects, maintaining and keeping models up to date becomes a critical best practice to ensure long-term success. Models must evolve together with the system and the organization's understanding of it. A model that is not actively maintained will quickly lose its relevance and usefulness, leading to a vicious cycle where outdated models are not used, and therefore, don't receive the resources needed to be kept current. To break this cycle, models should be treated as living artifacts that provide continuous value to stakeholders. The key to ensuring that models remain useful is to make them the main source for answering system-related questions, and to provide automation where possible. Automating data updates and exchanges can significantly reduce the manual effort needed to keep models accurate, especially when drawing from existing databases or external data sources. By integrating MBSE models with these external sources and automating data refreshes, the risk of inconsistencies is minimized, and the labor required to maintain the models is reduced. Where manual updates are necessary, creating user-friendly interfaces that allow domain experts (not just modeling experts) to input data ensures that the burden of model maintenance doesn't fall solely on a small team of modelers. This creates a virtuous cycle: the more models are used, the more they are maintained, which in turn increases their value, relevance, and use. By ensuring models stay current and relevant, they can continue to grow in value and serve as a reliable, central resource for the organization's decision-making processes. Models should be viewed and treated as key components of the digital engineering ecosystem, which requires similar rigor and management to vital software components that are responsible for providing and maintaining engineering data across the entire system life cycle [21][41][87][78].

- **Best Practice: Develop Model Libraries and Promote Modularity**

  A key best practice when scaling MBSE adoption is to start developing and utilizing libraries of reusable model elements. As organizations gain experience through pilot projects, the creation of libraries for interfaces, components, and other reusable system elements becomes essential for accelerating future projects. Establishing modular and reusable libraries like interface-, component-, or unit-libraries, enables teams to avoid starting from scratch for each project, streamlining model development and fostering consistency across efforts. Modular Open Systems Architecture (MOSA) principles can be employed to design major system interfaces that comply with widely supported standards, facilitating the reuse of requirements, designs, and test artifacts across multiple projects. These steps not only help with model partitioning and the reuse of elements, but also support and encourage broader MBSE adoption by providing ready to use, reference model elements that allow new projects to begin with a solid foundation. Over time, this practice enables teams to build models more efficiently, ensures consistency in system development, and reduces costs associated with recreating common elements. By making reusable models and modularity a cornerstone of the MBSE process, organizations can significantly accelerate project timelines and drive the successful scaling of MBSE across the enterprise [73][87][6][34][55][75][96].

- **Best Practice: Use of Documents in MBSE**

  Another best practice in the adoption of MBSE at this stage is to recognize that the need for readable documents alongside the transition to model-centric practices remains. It is important to acknowledge that not all stakeholders may engage with models directly from the beginning. Some disciplines might not be as involved in modeling or are slow in adopting new concepts of operation, but are reliant on the results. Therefore, there should be mechanisms in place to extract traditional systems engineering artifacts from models to accommodate those who are not accustomed to working directly with models. This visualization and conversion of modeled content into document- and presentation-reports are necessary to stay communicative to all teams and customers. Automated methods for document generation and integration from model-driven development play a critical role in this process, ensuring that design updates are consistently reflected in the generated documents. This dual approach allows organizations to leverage the advantages of MBSE while still meeting the traditional needs of stakeholders who may be less agile in adopting new paradigms. As companies transition to MBSE, maintaining the capability to produce communicative reports and documentation will facilitate smoother interactions with different domains, suppliers and partners that rely on traditional practices, ultimately leading to a less intrusive introduction of MBSE [80][6][34][84][97][91].

- **Best Practice: Build Flexibility into Models for Future Growth**

  As MBSE scales within an organization, it is critical to build flexibility into system models to accommodate future growth and new applications. Early investments in flexible model architectures can significantly reduce the accumulation of technical debt, ensuring that models remain adaptable and sustainable over time. This flexibility enables models to evolve alongside system requirements, preventing the need for major rework when expanding the model's scope or functionality. Experienced model architects should focus on balancing immediate needs with long-term adaptability, avoiding overly rigid designs that limit future use cases. By incorporating this flexibility up front, organizations can ensure that their MBSE approach remains responsive to changing demands, improving model usability, maintainability, and scalability over time [21][41][6][96].

Management

- **Best Practice: Make Models the Centerpiece Artifact**

  The final best practice at this stage is to commit to making the models the centerpiece artifact for referencing architecture, converting requirements into designs, and tracing verification and validation tasks throughout the design process. By prioritizing models, organizations ensure that changes begin with the model itself, reinforcing its role as the primary design reference. It is essential to allocate continuous resources for updating and sustaining these models, employing them as the foundation for design reviews and verifying requirements. With adequate training, stakeholders can conduct design reviews directly from the modeling tool, allowing them to witness firsthand how key elements of MBSE, such as reuse, modularity and encapsulation, enhance consistency, traceability and overall design quality. Regular model reviews should actively involve model contributors and domain experts, utilizing model files rather than static images to facilitate more effective discussions. While models serve as the main artifact, it can be a good practice to restrict access to certain parts of the model to prevent overwhelming users with excessive views and options that are only relevant for particular experts. Positioning models at the heart of the engineering workflow may require changes to organizational processes, but stepwise replacement of traditional activities and making sure that all stakeholder are aligned with the model-based approach is necessary for MBSE to succeed [39][18][6][55][57][91].

The maturity assessment outlined in this section provides a structured and comprehensive collection of challenges, pitfalls, and best practices. It offers high-level guidance designed to help organizations approach the adoption process in an informed and structured manner.

For a transition as complex and context-dependent as MBSE, providing precise, one-size-fits-all instructions is neither practical nor achievable. Each organization is unique, with distinct characteristics, goals, and constraints that only its management fully understands. Therefore, the assessment intentionally focuses on a holistic, high-level approach, enabling companies to identify critical areas for attention while ensuring that no key aspect of the adoption process is overlooked.

This approach empowers management to derive tailored, lower-level instructions and actionable steps specific to their organization's needs. For example, while the assessment emphasizes the importance of selecting tools that integrate effectively with existing infrastructure, the specific tool choice depends on the organization's unique technical and operational requirements. Similarly, while it highlights the need to address knowledge gaps and align divergent understandings of MBSE, the exact methods and content of training to achieve this, must be determined by the organization itself, as it has to fit their schedule and address their specific deficiencies.

To validate the structure and utility of the maturity assessment, we initiated an early evaluation effort. This process aims to gauge the MBSE community's agreement with the assessment's content and gather feedback to refine its structure where needed.

CHAPTER 6

Peer Review from MBSE Community

The development of a maturity assessment, as outlined in [69] and [70], typically involves a structured validation process consisting of multiple phases. This process begins with an early validation or proof of concept to ensure the initial design aligns with its intended purpose and resonates with the target audience. In later stages, case studies are often employed to test the practical application and evaluate the impact of the maturity assessment on real-world scenarios.

Given the scope and timeframe of this thesis, conducting an early evaluation phase was the most feasible approach. The aim was to gather feedback quickly and efficiently from a diverse group of MBSE professionals to assess whether the overall concept and structure of the maturity assessment were appropriate. This phase prioritized identifying any potential flaws or weaknesses in the design to ensure a solid foundation for future refinements and applications.

To achieve this, a survey was created using Google Forms and published in the LinkedIn Group (Name: *MBSE, Model Based Systems Engineering*), which serves as a community forum for professionals in the field of MBSE. The group can be accessed at: `https://www.linkedin.com/groups/4036633/`. The survey format was chosen to maximize accessibility and reach within the professional community, allowing participants to provide feedback at their convenience. While it was recognized that this method might not yield the depth of insight possible through dedicated expert interviews, it enabled a broader and more time-efficient collection of initial impressions and assessments.

6.1 Introduction to the Survey

Before starting the survey, participants were asked three preliminary questions to gather demographic and contextual information. These questions aimed to provide insights into the participants' age, their experience with MBSE and MBSE adoption efforts (via pre-defined options), and their years of experience in the field. All participants were at least 40 years old. Figure 6.1 and Figure 6.2 display the results of the MBSE experience questions, which reveal that most respondents have at least general experience using MBSE in projects, with two participants indicating extensive experience managing or consulting on MBSE adoption. Notably, seven out of the eight respondents reported more than four years of experience in MBSE or related areas.

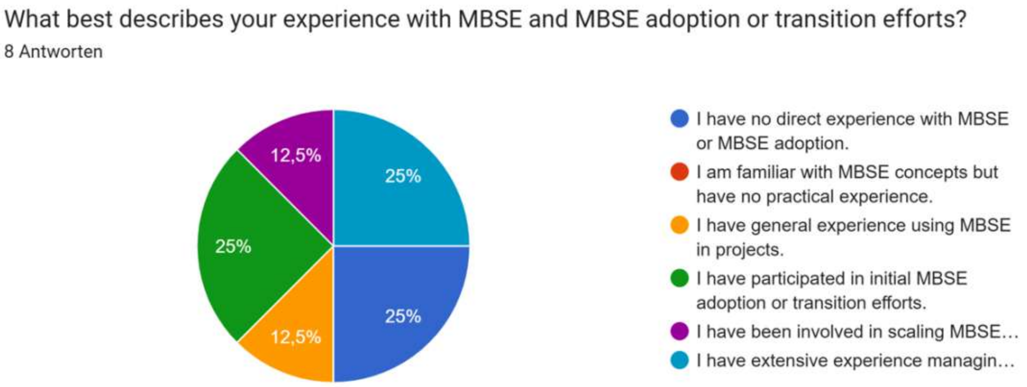


Figure 6.1: Experience with MBSE or MBSE adoption of the respondents

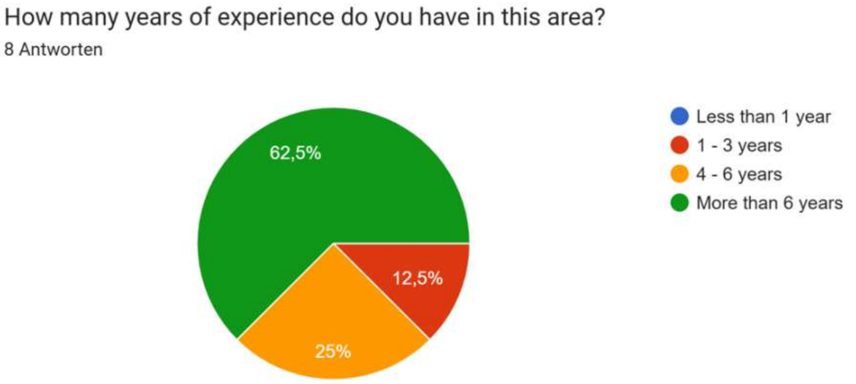


Figure 6.2: Years of experience in their respective field

While the majority of participants demonstrated significant expertise, a small discrepancy arose in the responses: two individuals selected "no direct experience with MBSE or MBSE adoption" in one question, despite no one reporting less than one year and only one reporting just one to three years of experience in the field. This could be attributed to differing interpretations of the question wording or their specific roles within MBSE-related contexts. Additionally, in the subsequent sections of the survey, two respondents mostly expressed more reservations or critical feedback regarding the maturity assessment. Although it is not possible to determine which participants provided this input, their responses offer valuable perspectives for refining the assessment and highlight the diversity of opinions within the sample.

This background information establishes a foundation for interpreting the survey results, demonstrating that the feedback predominantly comes from professionals with substantial MBSE experience, ensuring the insights are both informed and relevant.

6.2 Findings and Analysis

This section presents the results of the survey conducted to evaluate the maturity assessment. It covers respondents' feedback on the maturity levels, the elements within each level, the self-assessment questions, and the overall usability and applicability of the tool. The analysis highlights key trends, areas of agreement, and suggestions for improvement based on participant responses.

6.2.1 Agreement with Maturity Levels

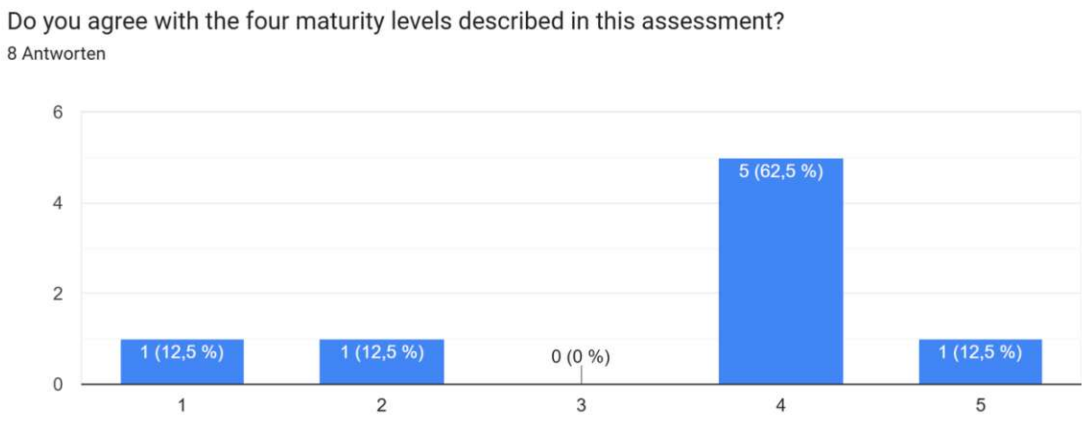


Figure 6.3: Agreement with Maturity Levels

The responses regarding agreement with the four maturity levels were generally positive, with most participants indicating alignment with the maturity assessment's depiction of MBSE adoption. However, two participants suggested that introducing an additional

Level 5 could enhance the clarity of the assessment. In their view, Level 4 would then represent adoption by some teams, while Level 5 would signify widespread adoption across most of the company. This adjustment would also shift pilot projects to a lower maturity level, better reflecting the progression toward full adoption. One of these participants also recommended providing more detailed descriptions of each level. For example, they suggested that Level 3 could describe MBSE use in specific, limited cases, such as modeling certain software aspects with some code generation, while Level 4 would involve full adoption, including modeling most software with a workflow that prioritizes models over code and incorporates substantial code generation. Level 5, according to this feedback, would emphasize the implementation of measurement practices and continual improvement.

6.2.2 Agreement with Elements at Each Level

Maturity Level	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Level 1	0	3	0	4	1
Level 2	0	0	1	6	1
Level 3	0	1	0	4	3
Level 4	0	2	1	2	3
Sum	0	6	2	16	8

Table 6.1: This table shows the agreement votings for each level and an additional sum of each voting for a good overview.

Table 6.1 shows clearly that the overall agreement throughout the levels is quite positive with 75% of all votes being "Agree" or "Strongly Agree". In the following, we will examine each level and its corresponding elements individually, incorporating relevant feedback provided by the respondents.

Level 1: Initial Preparation

The overall evaluation of Level 1 was generally positive, though it received the highest number of "Disagree" votes compared to other levels. Based on direct feedback and the individual ratings of its elements, this relatively critical view appears to stem from a few key points.

First, two respondents agreed that the emphasis on developing holistic systems thinking may be overstated. They argued that modeling is often domain-specific in practice and that focusing too much on holistic understanding could be perceived as overly theoretical. The takeaway here is to carefully consider which stakeholders genuinely benefit from holistic system understanding and for whom it may simply add unnecessary overhead.

Second, some feedback highlighted the desire to see tools and practical examples introduced earlier in the process. Respondents noted that good tools and simple yet practical examples can play a vital role in demonstrating the value of MBSE and how it could transform workflows. While selecting a final tool at this early stage is not advisable, starting with an easy-to-use tool can facilitate small pilot projects to help teams gain initial experience and confidence. However, respondents also cautioned against the potential pitfall of using poor examples—either overly simplistic "toy" examples or overly complex ones—which would fail to deliver the intended benefits and hinder early adoption efforts.

These points summarize the feedback received on Level 1 and highlight areas for potential refinement.

Level 2: Planning & Structure

The overall evaluation of Level 2 was positive, with 7 out of 8 respondents selecting "Agree" or "Strongly Agree". However, the direct feedback highlighted several areas for potential refinement.

One recurring theme in the feedback was the desire for a stronger focus on tools at this stage. Respondents emphasized that selecting tools with flexibility is crucial. Specifically, selecting tools that allow for easy replacement or migration without losing prior work.

Additionally, one respondent advocated for a bottom-up approach, arguing that adoption should start with compelling tools for developers rather than being imposed top-down by management. They reasoned that no amount of planning or evaluation can prevent developers from abandoning models and reverting to coding under time pressure, something frequently observed in projects, except in safety-critical systems where the dramatic cost of failure ensures model adherence. Early-stage "toy" projects were also recommended, as they provide an opportunity to explore different tools and options without prematurely committing to a specific infrastructure.

Another point raised was that the "tool access" element might be out of scope for Level 2, as it is a consideration relevant to all levels. Respondents suggested that tool access naturally evolves alongside the approach and the number of users involved. Furthermore, one respondent commented that issues with lacking IT infrastructure are less of a concern in modern contexts.

These insights suggest that while Level 2 is well-received overall, adjustments could further align it with practical considerations for tool selection and infrastructure at this stage.

Level 3: Pilot Projects

The overall rating for Level 3 was again very positive, with 7 out of 8 respondents selecting "Agree" or "Strongly Agree". However, the direct feedback offered some valuable insights for refinement.

One respondent highlighted the use of domain-specific languages (DSLs) in MBSE as a way to mitigate common pitfalls such as aimless modeling, overmodeling, or poor model architecture, issues more frequently encountered with general purpose languages like SysML. While this is a valid point and DSLs should be carefully considered during the tool and language selection process, it is important to recognize that DSLs and general-purpose modeling languages like SysML serve different purposes. DSLs excel in narrowly focused tasks, often avoiding overhead and inefficiencies, but they lack the ability to provide the holistic system view and broad understanding that SysML or similar languages can offer. For this reason, strategies have been included in the assessment to address and minimize these pitfalls when using general-purpose modeling languages.

Another key piece of feedback was the critique of pilot projects. Some respondents suggested that pilot projects are not always feasible or effective. Instead, they recommended exploring tools in real projects, supported by peer reviews, or applying MBSE to existing projects to document designs, improve maintainability, and enhance project sustainability. These decisions should always be guided by a cost/risk analysis to determine whether MBSE should be applied to new or existing projects.

This feedback reinforces points already covered in the thesis but highlights that the simplified descriptions provided in the maturity assessment might lack sufficient clarity. These descriptions should be improved in the next refinement step, to avoid ambiguity and better align with the underlying principles outlined in the thesis.

Level 4: Scaling MBSE Adoption

The distribution of feedback for Level 4 reflects a mix of opinions, with 2 votes for "Disagree", 1 for "Neutral", 2 for "Agree", and 3 for "Strongly Agree". This indicates that while the level was generally well-received, there are differing views on its implementation and focus.

One comment suggested that MBSE should not be enforced, allowing designers to choose the most efficient way to work. While this perspective is valid in certain use cases or specific project contexts, it does not align well with a company-wide MBSE adoption strategy. When an organization commits to transitioning to MBSE, the development lifecycle will eventually require full alignment. Resistance to change, often stemming from familiarity with existing workflows, may initially seem justified due to perceived efficiency. However, such resistance can hinder long-term progress, as adapting to the MBSE approach becomes necessary for a cohesive and scalable implementation across teams.

Another point raised by two respondents was the suggestion to introduce a fifth maturity level focused on continuous improvement. These respondents emphasized that pilot projects, which could begin as early as Level 1 or 2 with small "toy" examples, should be more distinct from the final maturity stage to better reflect the gradual scaling of MBSE efforts.

Lastly, one respondent noted that the assessment does not explicitly address whether organizations should adopt a single modeling language across all projects or allow for different languages depending on the diversity of processes within the organization. While this is a valid consideration, such decisions are highly context-dependent and should be explored during the tool and language selection process in Level 2. This step provides the flexibility to evaluate and choose the most suitable languages and tools for the organization's specific needs.

6.2.3 Feedback on Self-Assessment Questions

The feedback on the self-assessment questions was mixed, with 1 vote for "Disagree", 2 for "Neutral", 3 for "Agree", and 2 for "Strongly Agree". While the overall sentiment leans positive, some respondents felt that the holistic, company-wide adoption approach presented in the assessment might be overly broad.

Specifically, it was noted that MBSE adoption efforts often begin within smaller teams, which conduct their own evaluations and make localized decisions rather than following a top-down, organization-wide approach. This highlights a potential area for future research: comparing the effectiveness and challenges of the holistic, company-wide approach with a more team-focused, incremental approach to MBSE adoption. This differentiation could provide valuable insights into tailoring MBSE strategies for organizations at varying scales and levels of maturity.

6.2.4 Overall Agreement and Willingness to Use the Maturity Assessment

The final overall feedback on the maturity assessment was largely positive, as shown in Figure 6.4. On the second question, six respondents indicated they would consider using the maturity assessment for their job or company, while two voted "Strongly Disagree".

The direct feedback highlighted both the strengths and areas for improvement. One respondent referenced earlier suggestions, such as refining the assessment by adding a fifth level and addressing other points for clarity and usability. While one reiterated a preference for a more team-focused approach, as opposed to the current company-wide adoption framework. Notably, one respondent who disagreed with the current version provided their email for further collaboration, showing a willingness to contribute to its refinement.

On the positive side, the feedback also emphasized the relevance and value of the maturity assessment. Comments included praise for its applicability, with one respondent specifically mentioning its alignment with their current MBSE program and another calling the work "very important".

Overall, the assessment was well-received, with valuable feedback highlighting its potential impact while pointing out specific areas for refinement and future research.

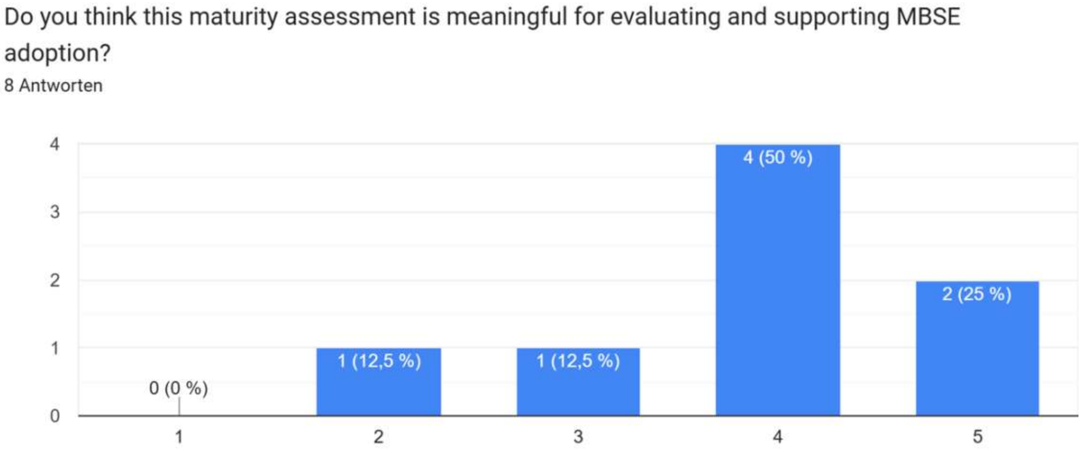


Figure 6.4: Overall maturity assessment rating

One additional critical perspective that was offered, suggested that the current version feels more like an adoption plan rather than a typical maturity assessment. This feedback reflects an important observation about the structure and purpose of the assessment. The research question guiding this work focused on translating insights into actionable guidance and advice, which resulted in a framework that emphasizes both progression and implementation. While this approach may align more closely with elements of an adoption plan, the maturity assessment structure was intentionally chosen to prioritize and organize the key elements into clear levels, providing a natural progression for organizations.

CHAPTER 7

Conclusion

As the complexity and technological demands of today's engineering landscape continue to escalate, the need for strategic approaches to manage increasingly complex, intelligent and global systems, that often exceed the ability of their designers to fully comprehend and control all aspects of the systems they are creating [89], becomes ever more relevant. MBSE offers a promising strategy to address these complexities, providing a pathway to manage the intricacies of future projects effectively.

This thesis contributes to this advancement by offering a structured approach to support companies in their transition to MBSE. The knowledge gathered and synthesized culminates in a maturity assessment designed to help practitioners focus their efforts, overcome challenges, avoid common pitfalls, and implement proven best practices.

Returning to the research questions of this work:

1. **RQ-1: What are the primary challenges encountered by companies in the adoption of Model-Based Systems Engineering (MBSE), as evidenced in existing literature?**

2. **RQ-2: What are the common pitfalls and best practices in the adoption of Model-Based Systems Engineering (MBSE), and how do these elements vary in priority and application across different organizational contexts?**

3. **RQ-3: To what extent can insights from past MBSE adoption efforts, including identified challenges, pitfalls, and best practices, be synthesized into a maturity assessment framework that guides organizations in the early stages of their MBSE transition by offering a structured adoption plan, and how is this framework received and validated by the MBSE community?**

The meta-synthesis approach outlined in this work addresses them as follows:

- **RQ-1** is answered by the challenges identified and systematically cataloged for each level of maturity, supported by references to the relevant literature.
- **RQ-2** is similarly addressed by the extensive list of pitfalls and best practices provided. While we found research showing that applicability does not vary among the best practices, only their prioritization can [7], we did not discover a comprehensive framework addressing prioritization. This thesis bridges the gap by offering a structured maturity assessment guiding practitioners throughout four stages.
- **RQ-3** is addressed through the development of the maturity assessment itself, which synthesizes lessons learned from past MBSE adoption efforts into a framework guiding organizations through the stages of their MBSE adoption. This approach balances the complexity of MBSE transitions with actionable, stage-specific steps, naturally prioritizing and focusing on key areas that past projects highlighted as critical. The overall reception of the MBSE community was positive with 75% of respondents agreeing that the assessment is meaningful for supporting MBSE adoption, while specific feedback was given on possible refinements and further research opportunities.

Ultimately, this work aims to reduce the daunting scale of MBSE transitions into manageable, cumulative steps, helping organizations allocate resources effectively and confidently progress toward sustainable implementation. While the journey toward a fully integrated MBSE approach is challenging and entails risks, the assessment equips organizations with a structured framework to tackle these complexities strategically.

To those still uncertain about the necessity or viability of transitioning to MBSE, the closing words of [71] serve as a, somewhat dramatic, yet fitting reminder:

> *"It is not necessary to change, survival is not mandatory."*

This quote highlights the urgency of addressing the growing complexities of modern systems. With no alternative strategy as effective, the choice becomes clear: embrace MBSE as a forward-looking solution. This work is intended to empower organizations to take that step confidently, armed with insights, structure, and guidance.

7.1 Contributions to Knowledge

This thesis makes significant contributions to the field of MBSE by providing a comprehensive and structured knowledge base on the adoption and transition process towards MBSE. Through the chosen meta-synthesis approach, grounded in a systematic literature review, it synthesizes scattered insights from existing research into a coherent framework that highlights the challenges, pitfalls, and best practices at different stages of the MBSE maturity journey.

The resulting maturity assessment tool offers a novel, high-level perspective that balances practical guidance with adaptability, addressing the inherent variability across organizations. This framework empowers organizations to tailor their transition strategies while ensuring no critical aspects are overlooked.

Additionally, this work addresses a gap in MBSE literature by explicitly focusing on the early adoption phases, where organizations often struggle the most. Offering a structured means for aligning efforts and resources.

7.2 Limitations and Future Work

While this thesis provides a comprehensive framework for guiding organizations through the MBSE adoption process, it is not without limitations. The meta-synthesis approach relies on existing literature, which inherently reflects the biases, scope, and limitations of previous studies. As such, while the findings are rooted in established research, they may not capture emerging practices or innovations in rapidly evolving industries.

Additionally, the high-level nature of the maturity assessment prioritizes generalizability over specificity. While this ensures broad applicability across diverse organizations, it leaves the task of deriving detailed, actionable steps to individual organizations. The absence of domain-specific guidance or low-level prescriptions may limit its direct applicability for companies seeking granular, step-by-step instructions tailored to their unique contexts.

Future research could focus on validating and refining the maturity assessment framework through empirical studies and real-world case applications. Testing the tool's effectiveness in diverse organizational settings would not only provide valuable feedback for improvement but also enhance its credibility and practical relevance.

The feedback gathered from the survey offers specific possibilities for refinement. For example, incorporating a fifth level into the framework and adjusting the focus of certain levels to align with earlier phases, as suggested by respondents, could better reflect the nuanced progression of MBSE adoption. Furthermore, exploring a more team-focused approach, as opposed to a holistic, company-wide adoption model, could uncover important differences in priorities and challenges faced by smaller teams versus entire organizations.

Additionally, in light of one respondent's observation that the current version feels more like an adoption plan rather than a typical maturity assessment, future research could aim to disentangle these two purposes. This might involve developing two distinct frameworks: one serving as a classic maturity assessment and the other as a general adoption plan. Such a separation would address varying user needs while ensuring clarity of purpose.

Finally, as MBSE and the fields around it continue to evolve, ongoing research should aim to keep the framework up-to-date. Incorporating advancements will ensure the maturity assessment remains a relevant and effective resource for organizations navigating MBSE transitions.

Overview of Generative AI Tools Used

During the development of this thesis, I utilized **ChatGPT**, an AI-powered language model, to enhance the clarity and structure of specific sentences and statements. This tool was instrumental in refining my writing, allowing me to better articulate complex ideas and ensure that my intended message was conveyed effectively when initial drafts fell short of achieving this. While all final content reflects my own thoughts and interpretations, the assistance provided by this generative AI tool contributed to a more polished presentation of the material.

Other Tools

Additionally, **Miro** was employed as a collaborative and visual brainstorming platform. It played a central role in conceptualizing, drafting, and structuring the maturity assessment framework. The tool facilitated the iterative development of the assessment and helped refine its visual and logical structure to support the goals of this research.

Furthermore, I occasionally used **DeepL** and **dict.cc** to verify specific terms or phrases and ensure accurate understanding of certain words or expressions in complex scientific papers. As English is not my first language, these tools provided supplementary support in clarifying technical or nuanced vocabulary.

List of Figures

Bibliography

[1] International Council on Systems Engineering (INCOSE), "Systems engineering vision 2020," *INCOSE-TP-2004-004-02*, vol. (version 2.03), 2007.

[2] B. Beihoff, C. Oster, S. Friedenthal, C. Paredis, D. Kemp, H. Stoewer, D. Nichols, and J. Wade, *A World in Motion – Systems Engineering Vision 2025*. 01 2014.

[3] L. Delligatti, *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Addison-Wesley Professional, 1st ed., 2013.

[4] T. A. McDermott, N. Hutchison, M. Clifford, E. Van Aken, A. Salado, and K. Henderson, "Benchmarking the benefits and current maturity of model-based systems engineering across the enterprise," *Systems Engineering Research Center (SERC)*, 2020.

[5] M. Chami and J.-M. Bruel, "A survey on mbse adoption challenges," in *INCOSE EMEA Sector Systems Engineering Conference (INCOSE EMEASEC 2018)*, (Berlin, Germany), pp. 1–16, November 2018.

[6] K. Henderson, T. McDermott, and A. Salado, "Mbse adoption experiences in organizations: Lessons learned," *Systems Engineering*, vol. 27, no. 1, pp. 214–239, 2024.

[7] C. Ploeg, K. Lai, and A. Olechowski, "Prioritization of best practices in the implementation of model-based systems engineering," *INCOSE International Symposium*, vol. 32, no. 1, pp. 961–975, 2022.

[8] S. Husung, C. Weber, A. Mahboob, and S. Kleiner, "Using model-based systems engineering for need-based and consistent support of the design process," *Proceedings of the Design Society*, vol. 1, p. 3369–3378, 2021.

[9] D. D. Walden, *Systems engineering handbook : a guide for system life cycle processes and activities*. Hoboken, NJ: John Wiley and Sons, fifth edition. ed., 2023.

[10] C. Tschirner, L. Bretz, R. Dumitrescu, and J. Gausemeier, "Applying model-based systems engineering for product engineering management concepts for industrial application," in *2015 IEEE International Symposium on Systems Engineering (ISSE)*, pp. 42–49, 2015.

[11] "ISO/IEC/IEEE International Standard - Systems and software engineering - System life cycle processes," *ISO/IEC/IEEE 15288:2023(E)*, pp. 1–128, 2023.

[12] H. Sillitto, D. Dori, R. M. Griego, S. Jackson, D. Krob, P. Godfrey, E. Arnold, J. N. Martin, and D. McKinney, "Defining "system": a comprehensive approach," *INCOSE International Symposium*, vol. 27, 2017.

[13] R. Jonkers and K. E. Shahroudi, "Connecting systems science, thinking and engineering to systems practice for managing complexity," in *65th Annual Proceedings for the International Society of the Systems Sciences*, vol. 65, 2021.

[14] "Systems Thinking - When it comes to solving complex problems, collaborating isn't enough." `https://systemsthinking.blog.gov.uk/2020/04/15/when-it-comes-to-solving-complex-problems-collaborating-isnt-enough/`. Accessed: 2024-06-10.

[15] H. Zhu and A. McDermott, "Optimal architecting strategy for partially developed products: Challenges and solutions," in *2023 IEEE International Systems Conference (SysCon)*, pp. 1–6, 2023.

[16] "Nasa - technical assessment - life-cycle reviews." `https://www.nasa.gov/reference/6-7-technical-assessment/`. Accessed: 2024-07-03.

[17] A. M. Madni and S. Purohit, "Economic analysis of model-based systems engineering," *Systems*, vol. 7, no. 1, 2019.

[18] D. R. Call and D. R. Herber, "Applicability of the diffusion of innovation theory to accelerate model-based systems engineering adoption," *Systems Engineering*, vol. 25, no. 6, pp. 574–583, 2022.

[19] E. Brusa, A. Calà, and D. Ferretto, "Systems, customer needs and requirements," in *Systems Engineering and Its Application to Industrial Product Development*, pp. 69–113, Springer International Publishing, 2018.

[20] T. Amorim, A. Vogelsang, and E. D. Canedo, "Decision support for process maturity improvement in model-based systems engineering," in *Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering*, (New York, NY, USA), p. 13–23, Association for Computing Machinery, 2022.

[21] R. A. Noguchi, "Recommended best practices based on mbse pilot projects," *INCOSE International Symposium*, vol. 29, no. 1, pp. 753–770, 2019.

[22] P. De Saqui-Sannes, R. A. Vingerhoeds, C. Garion, and X. Thirioux, "A taxonomy of mbse approaches by languages, tools and methods," *IEEE Access*, vol. 10, pp. 120936–120950, 2022.

[23] D. Bork and H. Fill, "Formal aspects of enterprise modeling methods: A comparison framework," in *47th Hawaii International Conference on System Sciences, HICSS 2014, Waikoloa, HI, USA, January 6-9, 2014*, pp. 3400–3409, IEEE Computer Society, 2014.

[24] T. Firchau, M. Jetzschmann, and F. Nohka, "Applying mbse to data handling system design of a small satellite platform for multiple missions," in *2022 IEEE Aerospace Conference (AERO)*, pp. 1–10, 2022.

[25] A. Akundi and V. Lopez, "A review on application of model based systems engineering to manufacturing and production engineering systems," *Procedia Computer Science*, vol. 185, pp. 101–108, 2021.

[26] Y. Guo, S. She, J. Zhao, and J. Guo, "Advantage analysis of mbse in the context of multi-historical development trends," in *Frontiers in Artifical Intelligence and Applications*, pp. 138–145, 12 2022.

[27] J. A. Estefan and T. Weilkiens, "Mbse methodologies," in *Handbook of Model-Based Systems Engineering*, (Cham), pp. 1–40, Springer International Publishing, 2020.

[28] M. Bouaicha, N. Machkour, I. E. Adraoui, and M. Zegrari, "Mbse grid: Operational analysis for the implementation of hydroelectric group health monitoring and management unit," in *Smart Applications and Data Analysis*, pp. 402–410, Springer International Publishing, 2022.

[29] C. Mandel, J. Kaspar, R. Heitmann, S. Horstmeyer, A. Martin, and A. Albers, "Implementation and assessment of a comprehensive model-based systems engineering methodology with regard to user acceptance in practice," *Procedia CIRP*, vol. 119, pp. 897–902, 2023. The 33rd CIRP Design Conference.

[30] T. Weilkiens, A. Scheithauer, M. Di Maio, and N. Klusmann, "Evaluating and comparing mbse methodologies for practitioners," in *2016 IEEE International Symposium on Systems Engineering (ISSE)*, pp. 1–8, 2016.

[31] M. Riesener, C. Doelle, S. Perau, P. Lossie, and G. Schuh, "Methodology for iterative system modeling in agile product development," *Procedia CIRP*, vol. 100, pp. 439–444, 2021. 31st CIRP Design Conference 2021 (CIRP Design 2021).

[32] C. Ponsard. and V. Ramon., "Applying and extending femmp to select an adequate mbse methodology," in *Proceedings of the 17th International Conference on Software Technologies - ICSOFT*, pp. 508–515, INSTICC, SciTePress, 2022.

[33] P. J. Younse, J. E. Cameron, and T. H. Bradley, "Comparative analysis of a model-based systems engineering approach to a traditional systems engineering approach for architecting a robotic space system through knowledge categorization," *Systems Engineering*, vol. 24, no. 3, pp. 177–199, 2021.

[34] J. L. Alvarez, H.-P. de Koning, D. Fischer, M. Wallum, H. Metselaar, and M. Kretzenbacher, "Best practices for model based systems engineering in esa projects," in *2018 AIAA SPACE and Astronautics Forum and Exposition*, 2018.

[35] "IEEE/ISO/IEC International Standard for Software, systems and enterprise - Architecture description," *ISO/IEC/IEEE 42010:2022(E)*, pp. 1–74, 2022.

[36] A. Khandoker, S. Sint, G. Gessl, K. Zeman, F. Jungreitmayr, H. Wahl, A. Wenigwieser, and R. Kretschmer, "Towards a logical framework for ideal mbse tool selection based on discipline specific requirements," *Journal of Systems and Software*, vol. 189, p. 111306, 2022.

[37] D. Bork, P. Langer, and T. Ortmayr, "A vision for flexible glsp-based web modeling tools," in *The Practice of Enterprise Modeling - 16th IFIP Working Conference, PoEM 2023, Vienna, Austria, November 28 - December 1, 2023, Proceedings* (J. P. A. Almeida, M. Kaczmarek-Heß, A. Koschmider, and H. A. Proper, eds.), vol. 497 of *Lecture Notes in Business Information Processing*, pp. 109–124, Springer, 2023.

[38] C. Ponsard and D. Darquennes, "Mbse tool selection process: Feedback from a railway case study," in *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 710–714, 2023.

[39] E. R. Carroll and R. J. Malins, "Systematic literature review: How is model-based systems engineering justified?," 3 2016.

[40] A. Vogelsang, T. Amorim, F. Pudlitz, P. Gersing, and J. Philipps, "Should i stay or should i go? on forces that drive and prevent mbse adoption in the embedded systems industry," 2017.

[41] R. A. Noguchi, R. J. Minnichelli, and M. J. Wheaton, "Architecting success in model based systems engineering pilot projects," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 755–760, 2019.

[42] G. Wang, "Implementing a model-based, digital engineering enterprise for a defense systems integrator – an ongoing journey," *INCOSE International Symposium*, vol. 30, no. 1, pp. 783–798, 2020.

[43] V. Mittal and S. Gillespie, "Using model-based systems engineering to avoid unnecessary technology resulting from dynamic requirements," *IEEE Transactions on Engineering Management*, vol. 71, pp. 2660–2671, 2024.

[44] G. Dubos, S. Schreiner, D. A. Wagner, G. Jones, A. A. Kerzhner, and J. Kaderka, "Architecture modeling on the europa project," in *AIAA SPACE 2016*, 2016.

[45] J. Baschin, R. Schmidt, D. Schneider, T. Vietor, and U. V. Kizgin, "Linking cross-domain information to support the development of complex systems," *Proceedings of the Design Society*, vol. 3, p. 2455–2464, 2023.

[46] H. Gustavsson, E. P. Enoiu, and J. Carlson, “Model-based system engineering adoption in the vehicular systems domain,” in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pp. 907–911, 2022.

[47] L. Boggero, P. D. Ciampa, and B. Nagel, “The application of the agile 4.0 mbse architectural framework for the modeling of system stakeholders, needs and requirements,” in *ICAS Conference 2021*, 2021.

[48] P. Younse, J. Cameron, and T. H. Bradley, “Comparative analysis of model-based and traditional systems engineering approaches for simulating a robotic space system architecture through automatic knowledge processing,” *Systems Engineering*, vol. 25, no. 4, pp. 360–386, 2022.

[49] T. Huldt and I. Stenius, “State-of-practice survey of model-based systems engineering,” *Systems Engineering*, vol. 22, no. 2, pp. 134–145, 2019.

[50] J. Wade, “Systems engineering transformation through digital engineering,” in *Systems Engineering for the Digital Age*, pp. 111–136, John Wiley and Sons, Ltd, 2023.

[51] E. B. Rogers III and S. W. Mitchell, “Mbse delivers significant return on investment in evolutionary development of complex sos,” *Systems Engineering*, vol. 24, no. 6, pp. 385–408, 2021.

[52] K. Henderson and A. Salado, “Value and benefits of model-based systems engineering (mbse): Evidence from the literature,” *Systems Engineering*, vol. 24, no. 1, pp. 51–66, 2021.

[53] K. X. Campo, T. Teper, C. E. Eaton, A. M. Shipman, G. Bhatia, and B. Mesmer, “Model-based systems engineering: Evaluating perceived value, metrics, and evidence through literature,” *Systems Engineering*, vol. 26, no. 1, pp. 104–129, 2023.

[54] K. Henderson, T. McDermott, E. Van Aken, and A. Salado, “Towards developing metrics to evaluate digital engineering,” *Systems Engineering*, vol. 26, no. 1, pp. 3–31, 2023.

[55] T. McDermott, K. Henderson, E. Van Aken, A. Salado, and J. Bradley, “Measuring the roi of digital engineering: It’s a journey, not a number,” *AIRC Perspectives*, pp. 1–8, 2022.

[56] J. M. C. Oh, M. G. Fifield, S. F. Scandore, I. A. Trettel, and B. Donitz, “Model-based systems engineering efficiencies,” in *2023 IEEE Aerospace Conference*, pp. 1–7, 2023.

[57] S. W. Mitchell, “Transitioning the swfts program combat system product family from traditional document-centric to model-based systems engineering,” *Systems Engineering*, vol. 17, no. 3, pp. 313–329, 2014.

[58] P. Assef and J. Geiger, “Adoption of model-based systems engineering in traditional dod systems,” *Defense Acquisition Research Journal*, vol. 30, pp. 46–73, 04 2023.

[59] A. Akundi, W. Ankobiah, O. Mondragon, and S. Luna, “Perceptions and the extent of model-based systems engineering (mbse) use – an industry survey,” in *2022 IEEE International Systems Conference (SysCon)*, pp. 1–7, 2022.

[60] M. Schöberl, E. Rebentisch, J. Trauer, M. Mörtl, and J. Fottner, “Evaluating mbse potential using product and development characteristics – a statistical investigation,” *Proceedings of the Design Society: DESIGN Conference*, vol. 1, p. 2385–2394, 2020.

[61] A. P. Siddaway, A. M. Wood, and L. V. Hedges, “How to do a systematic review: A best practice guide for conducting and reporting narrative reviews, meta-analyses, and meta-syntheses,” *Annual Review of Psychology*, vol. 70, no. 1, pp. 747–770, 2019.

[62] C. Hoon, “Meta-synthesis of qualitative case studies: An approach to theory building,” *Organizational Research Methods*, vol. 16, no. 4, pp. 522–556, 2013.

[63] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” vol. 2, 01 2007.

[64] C. Hansen, H. Steinmetz, and J. Block, “How to conduct a meta-analysis in eight steps: a practical guide,” *Management Review Quarterly*, vol. 72, 11 2021.

[65] “Scopus Keyword Search: mbse.” `https://www.scopus.com/results/results.uri?sort=plf-f&src=s&sid=4864d821ae9097f9eb74bd6587319f7f&sot=a&sdt=b&sl=312&s=ALL%28mbse%29&origin=searchbasic&editSaveSearch=&txGid=51ec7bb16ffbac478d4981919744c00c&sessionSearchId=4864d821ae9097f9eb74bd6587319f7f&limit=10`. Accessed: 2024-05-06.

[66] “IEEE Keyword Search: mbse.” `https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=MBSE`. Accessed: 2024-05-06.

[67] J. Paul and M. Barari, “Meta-analysis and traditional systematic literature reviews—what, why, when, where, and how?,” *Psychology and Marketing*, vol. 39, 03 2022.

[68] “Reference Management Software: zotero.” `https://www.zotero.org/`. Accessed: 2024-05-28.

[69] T. de Bruin, M. Rosemann, R. Freeze, and U. Kulkarni, “Understanding the main phases of developing a maturity assessment model,” in *ACIS 2005 Proceedings - 16th Australasian Conference on Information Systems*, 2005.

[70] R. Knackstedt, J. Pöppelbuß, and J. Becker, “Vorgehensmodell zur entwicklung von reifegradmodellen,” in *Wirtschaftsinformatik*, 2009.

[71] INCOSE MBSE Initiative, “Systems engineering your mbse implementation: Where are you on your mbse journey?,” *INCOSE International Symposium*, vol. 32, no. 1, pp. 249–261, 2022.

[72] K. Henderson and A. Salado, “The effects of organizational structure on mbse adoption in industry: Insights from practitioners,” *Engineering Management Journal*, vol. 36, no. 1, pp. 117–143, 2024.

[73] M. Chami, A. Aleksandraviciene, A. Morkevicius, and J.-M. Bruel, “Towards solving mbse adoption challenges: The d3 mbse adoption toolbox,” *INCOSE International Symposium*, vol. 28, no. 1, pp. 1463–1477, 2018.

[74] M. Pratt and M. Dabkowski, “Analyzing the integration of mbse approaches within the aerospace industry according to utaut,” *Industrial and Systems Engineering Review*, vol. 10, pp. 127–134, Dec. 2022.

[75] L. Wang, M. Izygon, S. Okon, H. Wagner, and L. Garner, “Effort to accelerate mbse adoption and usage at jsc,” in *AIAA SPACE 2016*, 2016.

[76] A. Akundi and W. Ankobiah, “Mapping industry workforce needs to academic curricula – a workforce development effort in model-based systems engineering,” *Systems Engineering*, vol. 27, no. 4, pp. 685–698, 2024.

[77] T. Schumacher, T. Ammersdörfer, and D. Inkermann, “Development and application of simulation games to introduce model-based systems engineering,” in *Towards a new future in engineering education, new scenarios that european alliances of tech universities open up*, p. 702–709, Universitat Politècnica de Catalunya, Sep 2022.

[78] T. McDermott, K. Henderson, E. Van Aken, and A. Salado, “Framework for and progress of adoption of digital and model-based systems engineering into engineering enterprises,” in *The Proceedings of the 2023 Conference on Systems Engineering Research*, pp. 69–82, Springer Nature Switzerland, 2024.

[79] S. Dawson and A. Batchelor, “Empowering engineers in a digital engineering transition: Applying organizational psychology and systems thinking approaches to define the problem and to develop recommended actions,” *INCOSE International Symposium*, vol. 32, no. 1, pp. 594–607, 2022.

[80] J. Hallqvist and J. Larsson, “Introducing mbse by using systems engineering principles,” *INCOSE International Symposium*, vol. 26, no. 1, pp. 512–525, 2016.

[81] T. Amorim, A. Vogelsang, F. Pudlitz, P. Gersing, and J. Philipps, “Strategies and best practices for model-based systems engineering adoption in embedded systems industry,” in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 203–212, 2019.

[82] J. G. Nanfuka and R. Oosthuizen, "Mbse-lite: A framework for adopting model-based systems engineering in small and medium-sized enterprises in south africa," *Business, Management and Economics Engineering*, vol. 21, pp. 218–236, 2023.

[83] R. Faudou and J.-M. Bruel, "An industrial feedback on model-based requirements engineering in systems engineering context," in *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, pp. 190–199, 2016.

[84] O. Sindiy, T. Mozafari, and C. Budney, "Application of model-based systems engineering for the development of the asteroid redirect robotic mission," in *AIAA SPACE 2016*, 2016.

[85] B. L. Papke, G. Wang, R. Kratzke, and C. Schreiber, "Implementing mbse – an enterprise approach to an enterprise problem," *INCOSE International Symposium*, vol. 30, no. 1, pp. 1550–1567, 2020.

[86] S. Bonnet, J.-L. Voirin, V. Normand, and D. Exertier, "Implementing the mbse cultural change: Organization, coaching and lessons learned," *INCOSE International Symposium*, vol. 25, no. 1, pp. 508–523, 2015.

[87] B. Papke, M. Hause, D. Hetherington, S. McGervey, and S. Rodriguez, "Mbse model management pain points – wait, this looks familiar!," *INCOSE International Symposium*, vol. 33, no. 1, pp. 273–289, 2023.

[88] J. Suryadevara and S. Tiwari, "Adopting mbse in construction equipment industry: An experience report," in *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 512–521, 2018.

[89] T. Weilkiens, "Adoption of mbse in an organization," in *Handbook of Model-Based Systems Engineering* (A. M. Madni, N. Augustine, and M. Sievers, eds.), pp. 1–19, Springer International Publishing, 2020.

[90] J. B. Holladay, J. Knizhnik, K. J. Weiland, A. Stein, T. Sanders, and P. Schwindt, "Mbse infusion and modernization initiative (miami): "hot" benefits for real nasa applications," in *2019 IEEE Aerospace Conference*, pp. 1–14, 2019.

[91] A. Vasenev, W. T. Suermondt, A. Behl, and J. Lukkien, "Step-wise mbse introduction into a company: An interface-centric case study," in *2023 18th Annual System of Systems Engineering Conference (SoSe)*, pp. 1–6, 2023.

[92] T. Li, L. Pollettini Marcos, W. Huang, C. R. Kenley, K. A. Douglas, E. A. Madsen, and A. W. Fentiman, "Learning mbse online: A tale of two professional cohorts," *Systems*, vol. 11, no. 5, 2023.

[93] E. Mancin, "How model based systems engineering streamlines the development of complex systems," in *INCOSE Italia Conference on Systems Engineering*, 2014.

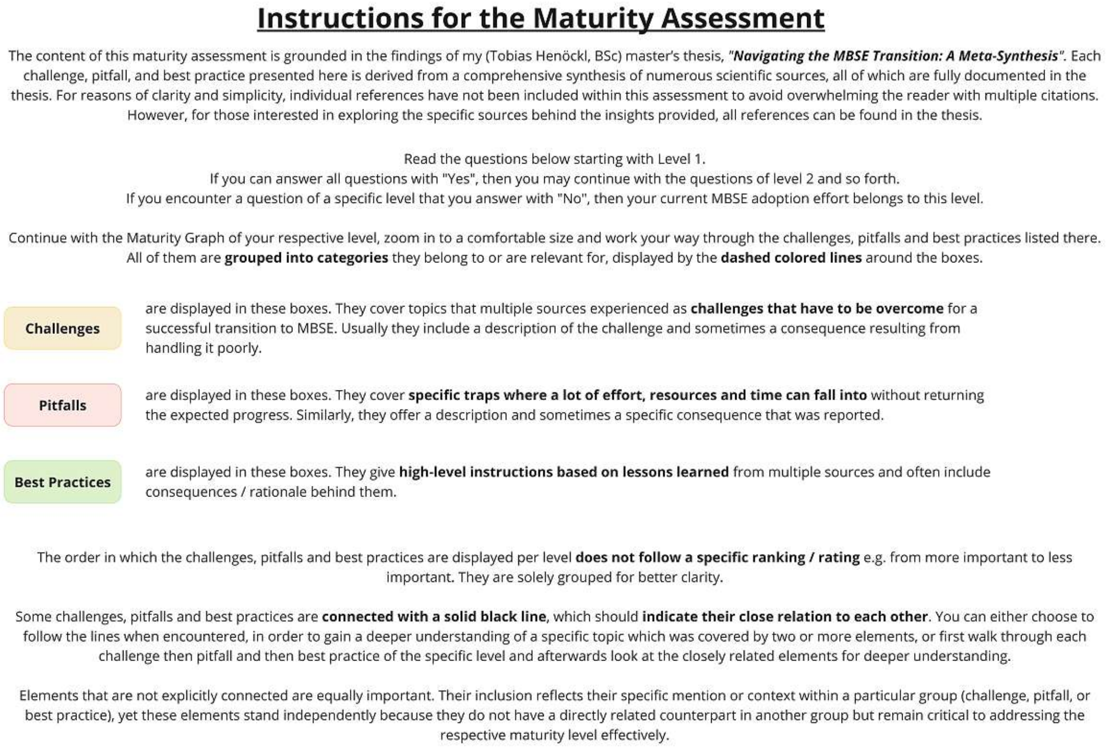
[94] J.-P. LERAT, “5.5.2 three reasons why document-based se (usually) works better than (most of) mbse,” *INCOSE International Symposium*, vol. 20, no. 1, pp. 723–738, 2010.

[95] M. Orosz, B. Duffy, C. Charlton, H. Saunders, and E. Thomas, “Unique challenges in mission engineering and technology integration,” in *Systems Engineering for the Digital Age*, pp. 665–681, John Wiley and Sons, Ltd, 2023.

[96] B. Papke, M. Hause, and D. Hetherington, “Fuse agility as a foundation for sound mbse lifecycle management,” *INSIGHT*, vol. 26, no. 2, pp. 34–38, 2023.

[97] R. Ferguson, J. Marshall, and M. Guzman, “Adapting progressive mbse development to document based contracts,” in *2020 IEEE Aerospace Conference*, pp. 1–11, 2020.

Appendix



Instructions for the Maturity Assessment

The content of this maturity assessment is grounded in the findings of my (Tobias Henöckl, BSc) master's thesis, ***"Navigating the MBSE Transition: A Meta-Synthesis"***. Each challenge, pitfall, and best practice presented here is derived from a comprehensive synthesis of numerous scientific sources, all of which are fully documented in the thesis. For reasons of clarity and simplicity, individual references have not been included within this assessment to avoid overwhelming the reader with multiple citations. However, for those interested in exploring the specific sources behind the insights provided, all references can be found in the thesis.

Read the questions below starting with Level 1.
If you can answer all questions with "Yes", then you may continue with the questions of level 2 and so forth.
If you encounter a question of a specific level that you answer with "No", then your current MBSE adoption effort belongs to this level.

Continue with the Maturity Graph of your respective level, zoom in to a comfortable size and work your way through the challenges, pitfalls and best practices listed there. All of them are **grouped into categories** they belong to or are relevant for, displayed by the **dashed colored lines** around the boxes.

Challenges are displayed in these boxes. They cover topics that multiple sources experienced as **challenges that have to be overcome** for a successful transition to MBSE. Usually they include a description of the challenge and sometimes a consequence resulting from handling it poorly.

Pitfalls are displayed in these boxes. They cover **specific traps where a lot of effort, resources and time can fall into** without returning the expected progress. Similarly, they offer a description and sometimes a specific consequence that was reported.

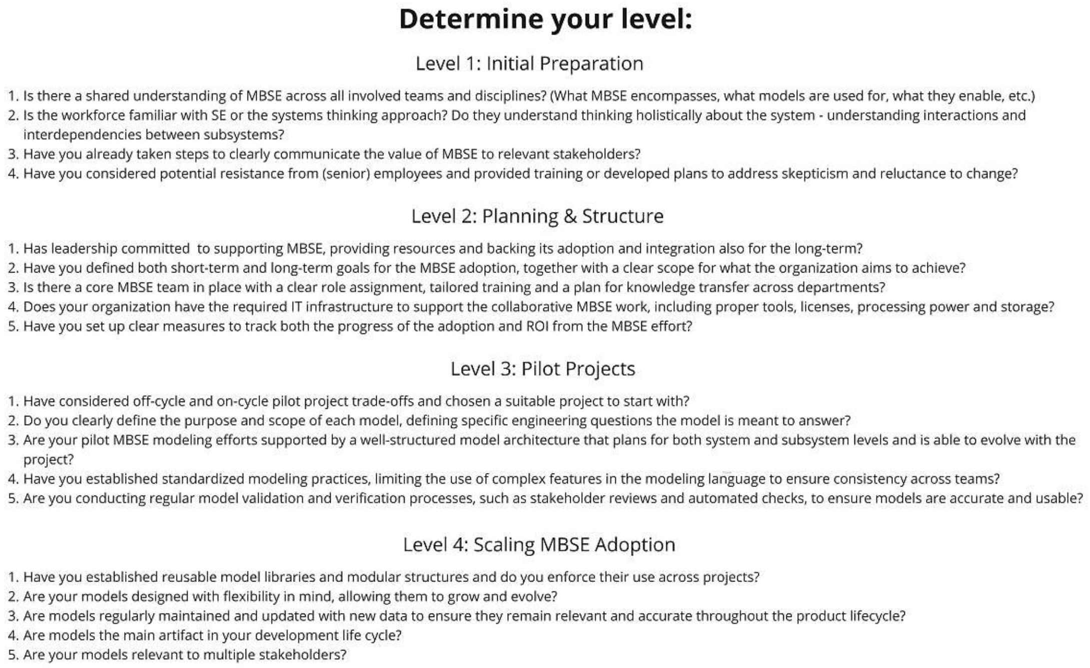
Best Practices are displayed in these boxes. They give **high-level instructions based on lessons learned** from multiple sources and often include consequences / rationale behind them.

The order in which the challenges, pitfalls and best practices are displayed per level **does not follow a specific ranking / rating** e.g. from more important to less important. They are solely grouped for better clarity.

Some challenges, pitfalls and best practices are **connected with a solid black line**, which should **indicate their close relation to each other**. You can either choose to follow the lines when encountered, in order to gain a deeper understanding of a specific topic which was covered by two or more elements, or first walk through each challenge then pitfall and then best practice of the specific level and afterwards look at the closely related elements for deeper understanding.

Elements that are not explicitly connected are equally important. Their inclusion reflects their specific mention or context within a particular group (challenge, pitfall, or best practice), yet these elements stand independently because they do not have a directly related counterpart in another group but remain critical to addressing the respective maturity level effectively.

Figure 1: MBSE Maturity Assessment - Instructions



Determine your level:

Level 1: Initial Preparation

1. Is there a shared understanding of MBSE across all involved teams and disciplines? (What MBSE encompasses, what models are used for, what they enable, etc.)
2. Is the workforce familiar with SE or the systems thinking approach? Do they understand thinking holistically about the system - understanding interactions and interdependencies between subsystems?
3. Have you already taken steps to clearly communicate the value of MBSE to relevant stakeholders?
4. Have you considered potential resistance from (senior) employees and provided training or developed plans to address skepticism and reluctance to change?

Level 2: Planning & Structure

1. Has leadership committed to supporting MBSE, providing resources and backing its adoption and integration also for the long-term?
2. Have you defined both short-term and long-term goals for the MBSE adoption, together with a clear scope for what the organization aims to achieve?
3. Is there a core MBSE team in place with a clear role assignment, tailored training and a plan for knowledge transfer across departments?
4. Does your organization have the required IT infrastructure to support the collaborative MBSE work, including proper tools, licenses, processing power and storage?
5. Have you set up clear measures to track both the progress of the adoption and ROI from the MBSE effort?

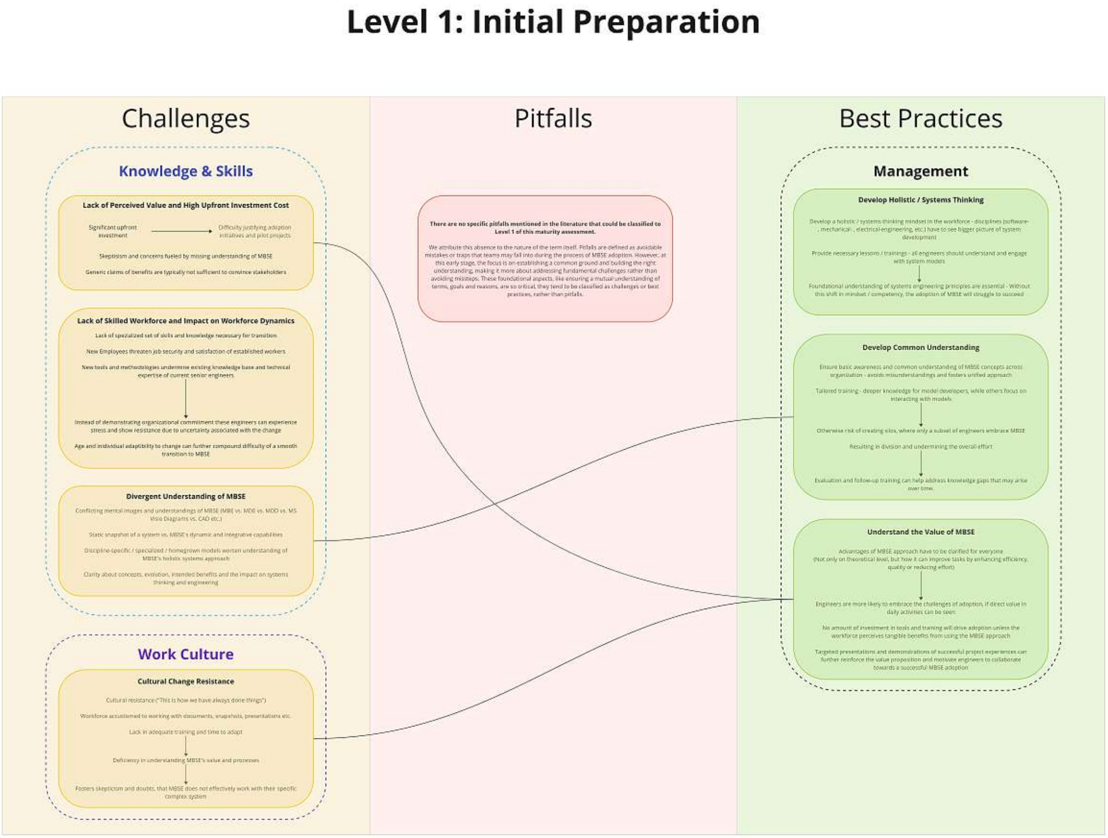
Level 3: Pilot Projects

1. Have considered off-cycle and on-cycle pilot project trade-offs and chosen a suitable project to start with?
2. Do you clearly define the purpose and scope of each model, defining specific engineering questions the model is meant to answer?
3. Are your pilot MBSE modeling efforts supported by a well-structured model architecture that plans for both system and subsystem levels and is able to evolve with the project?
4. Have you established standardized modeling practices, limiting the use of complex features in the modeling language to ensure consistency across teams?
5. Are you conducting regular model validation and verification processes, such as stakeholder reviews and automated checks, to ensure models are accurate and usable?

Level 4: Scaling MBSE Adoption

1. Have you established reusable model libraries and modular structures and do you enforce their use across projects?
2. Are your models designed with flexibility in mind, allowing them to grow and evolve?
3. Are models regularly maintained and updated with new data to ensure they remain relevant and accurate throughout the product lifecycle?
4. Are models the main artifact in your development life cycle?
5. Are your models relevant to multiple stakeholders?

Figure 2: MBSE Maturity Assessment - Questions



Lack of Perceived Value and High Upfront Investment Cost

Significant upfront investment → Difficulty justifying adoption initiatives and pilot projects

Skeptisism and concerns fueled by missing understanding of MBSE

Generic claims of benefits are typically not sufficient to convince stakeholders

Lack of Skilled Workforce and Impact on Workforce Dynamics

Lack of specialized set of skills and knowledge necessary for transition

New Employees threaten job security and satisfaction of established workers

New tools and methodologies undermine existing knowledge base and technical expertise of current senior engineers

Instead of demonstrating organizational commitment these engineers can experience stress and show resistance due to uncertainty associated with the change

Age and individual adaptibility to change can further compound difficulty of a smooth transition to MBSE

Divergent Understanding of MBSE

Conflicting mental images and understandings of MBSE (MBE vs. MDE vs. MDD vs. MS Visio Diagrams vs. CAD etc.)

Static snapshot of a system vs. MBSE's dynamic and integrative capabilities

Discipline-specific / specialized / homegrown models worsen understanding of MBSE's holistic systems approach

Clarity about concepts, evolution, intended benefits and the impact on systems thinking and engineering

Cultural Change Resistance

Cultural resistance ("This is how we have always done things")

Workforce accustomed to working with documents, snapshots, presentations etc.

Lack in adequate training and time to adapt

Deficiency in understanding MBSE's value and processes

Fosters skepticism and doubts, that MBSE does not effectively work with their specific complex system

There are no specific pitfalls mentioned in the literature that could be classified to Level 1 of this maturity assessment.

We attribute this absence to the nature of the term itself. Pitfalls are defined as avoidable mistakes or traps that teams may fall into during the process of MBSE adoption. However, at this early stage, the focus is on establishing a common ground and building the right understanding, making it more about addressing fundamental challenges rather than avoiding missteps. These foundational aspects, like ensuring a mutual understanding of terms, goals and reasons, are so critical, they tend to be classified as challenges or best practices, rather than pitfalls.

Develop Holistic / Systems Thinking

Develop a holistic / systems thinking mindset in the workforce - disciplines (software-, mechanical- , electrical-engineering, etc.) have to see bigger picture of system development

Provide necessary lessons / trainings - all engineers should understand and engage with system models

Foundational understanding of systems engineering principles are essential - Without this shift in mindset / competency, the adoption of MBSE will struggle to succeed

Develop Common Understanding

Ensure basic awareness and common understanding of MBSE concepts across organization - avoids misunderstandings and fosters unified approach

Tailored training - deeper knowledge for model developers, while others focus on interacting with models

Otherwise risk of creating silos, where only a subset of engineers embrace MBSE

Resulting in division and undermining the overall effort

Evaluation and follow-up training can help address knowledge gaps that may arise over time.

Understand the Value of MBSE

Advantages of MBSE approach have to be clarified for everyone
(Not only on theoretical level, but how it can improve tasks by enhancing efficiency, quality or reducing effort)

Engineers are more likely to embrace the challenges of adoption, if direct value in daily activities can be seen.

No amount of investment in tools and training will drive adoption unless the workforce perceives tangible benefits from using the MBSE approach

Targeted presentations and demonstrations of successful project experiences can further reinforce the value proposition and motivate engineers to collaborate towards a successful MBSE adoption

Figure 3: MBSE Maturity Assessment - Level 1: Initial Preparation

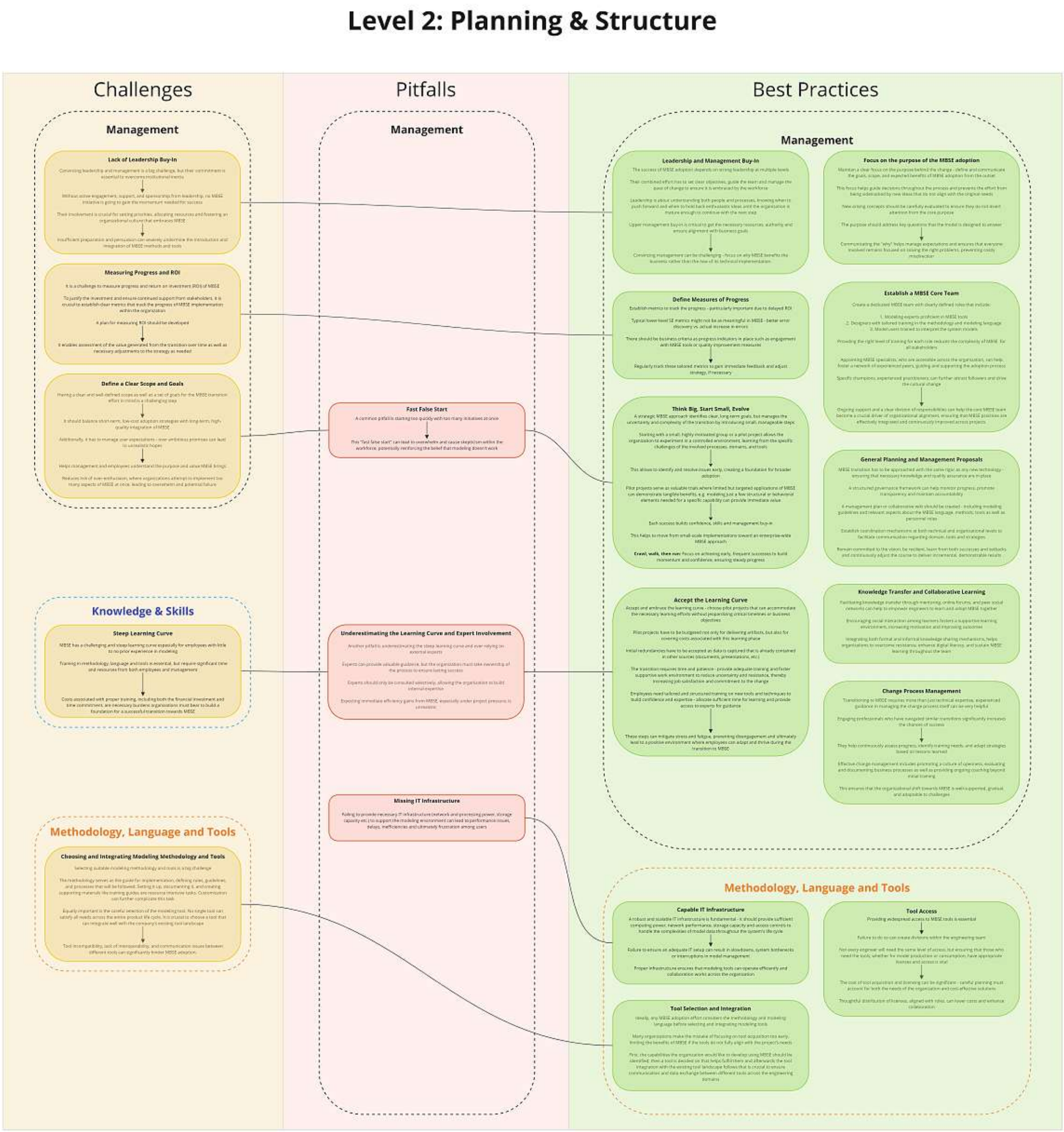


Figure 4: MBSE Maturity Assessment - Level 2: Planning & Structure

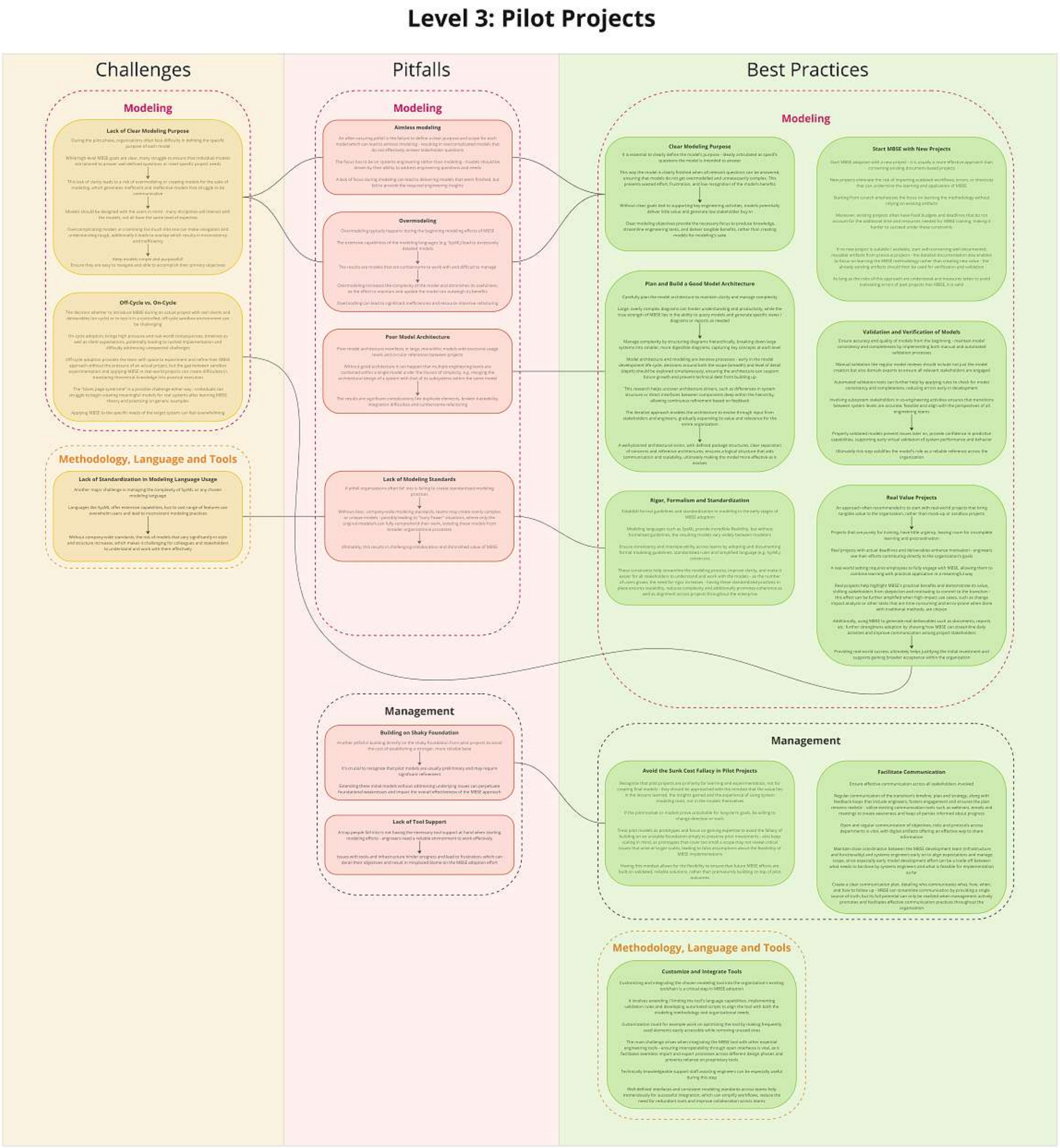


Figure 5: MBSE Maturity Assessment - Level 3: Pilot Projects

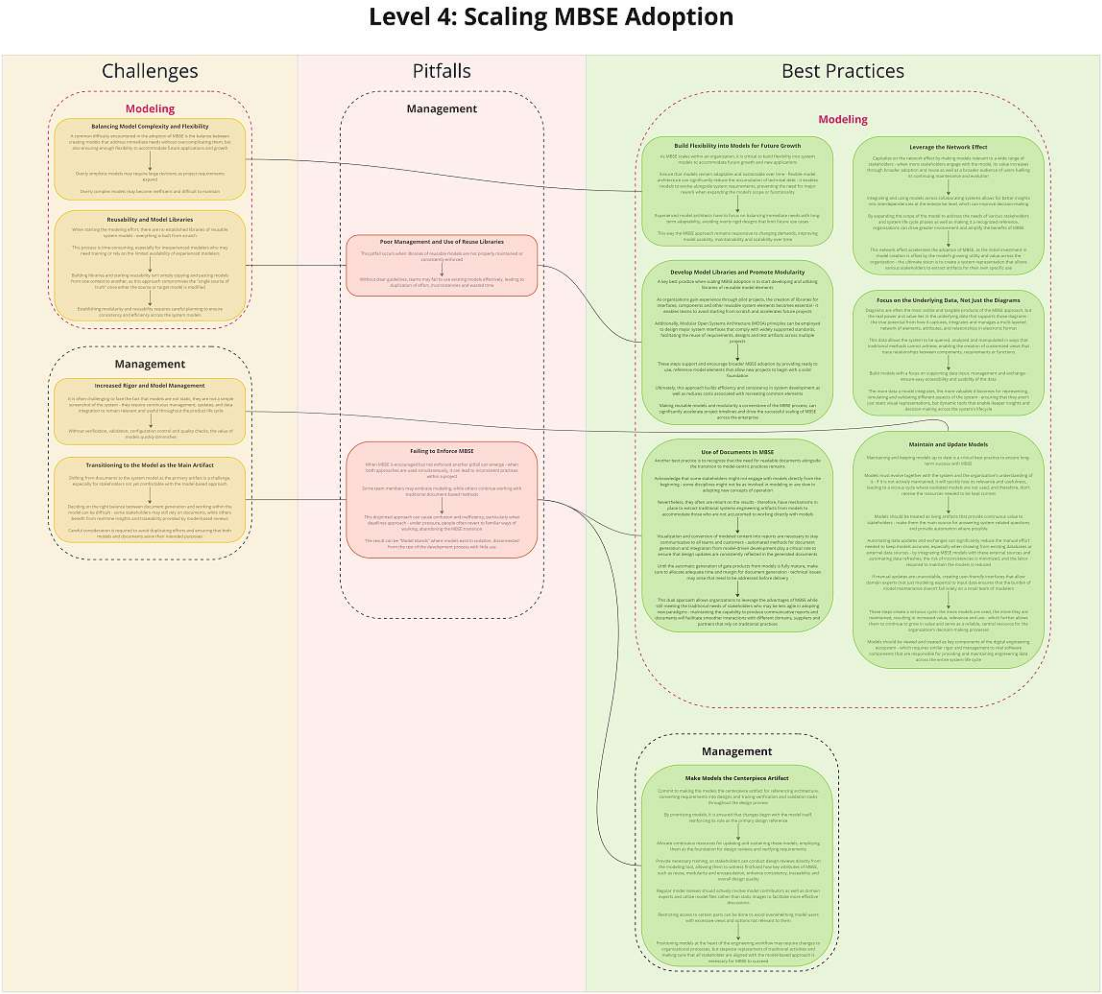


Figure 6: MBSE Maturity Assessment - Level 4: Scaling MBSE Adoption