Implementing Systems Engineering in New Product Development at Benchmark Electronics

David de Groot

MSc Thesis 15-Mar-23

University of Twente

Benchmark Electronics

Public version

Implementing Systems Engineering in New Product Development at Benchmark Electronics
MSc Thesis
David de Groot – s1955152

Faculty of Engineering Technology
Department of Design, Production and Management

Master programme: Industrial Design Engineering
Master track: Management of Product Design

Educational Institution
University of Twente
Drienerlolaan 5
7522 NB Enschede

Company
Benchmark Electronics
Lelyweg 10
7602 EA Almelo

Examination date
March 30, 2023

Examination committee
Prof.Dr. I. Gibson (chair)
Ir. I.F. Lutters – Weustink (supervisor)
Dr.Ir. K. Nizamis (external member)
J. Nijeboer (company supervisor)

This is the public version of this thesis. In some areas of the thesis, confidential information is withheld without mention. For the complete reasoning and results, request access and consult the confidential version.

FOREWORD

I am pleased to present this thesis on the design of a new implementation for Systems Engineering (SE) at Benchmark Almelo. For the past 2.5 years, I have been working at Benchmark, a company that designs and manufactures products in the consumer, industrial, and medical fields. During this time, I have had the opportunity to gain valuable experience and develop my skills as an engineer. It is with great excitement that I now present this thesis, which represents the culmination of my work at Benchmark.

This thesis is about SE, a discipline that involves the design, development, and management of complex systems. It is vital for creating high-quality products that meet the needs of customers and stakeholders. What motivates me about SE is that, with proper structure and teamwork, it makes it possible to create amazing feats of technology. It is this belief that has driven me throughout the research and writing of this thesis.

Benchmark is a company that values innovation and excellence. The products that we design have the potential to save lives, improve quality of life, or enhance productivity. It is a great honour to be a part of this mission, and I am proud to contribute to the company's way of working with this publication. In this thesis, I describe a new implementation for SE that is tailored to the needs of Benchmark. I believe that this implementation has the potential to streamline our processes, increase efficiency, and ultimately lead to better products.

I would like to take this opportunity to thank my supervisor at Benchmark, Jan Nijeboer, for his guidance, support, and encouragement throughout this project. Next, Henry Kompagnie, for entrusting me with reworking his unfinished SE process draft and his unwavering support and positivity therein. This project would not have been possible without the support and collaboration of the entire Benchmark team. I would also like to thank all of my colleagues at Benchmark for their contributions, feedback, and insights.

I would also like to thank my supervisor from the University of Twente, Ilanit Lutters, for her constructive feedback and guidance. Her insights and expertise have been invaluable throughout this project, and kept me on track. Finally, I would like to express my gratitude to my family and friends for their unwavering support and encouragement. Their love and encouragement have been a constant source of inspiration for me.

In conclusion, I hope that this thesis will serve as a valuable resource for Benchmark (and for others who are interested in SE). I believe that the implementation described in this thesis has the potential to positively impact the way that we design and develop products, and I am excited to see the impact that it will have.

ABSTRACT

This master thesis explores the benefits of Systems Engineering (SE) for Benchmark Electronics and aims to design a comprehensive SE program for Benchmark Almelo. SE is a multidisciplinary approach to designing and managing complex systems that incorporates various engineering, management, and organizational processes, and is becoming increasingly important in the high tech sector. The thesis addresses the main research question of how SE can be effectively implemented in new product development at Benchmark, and presents a SE program that is compatible with the company's existing processes and adapted to its project life cycle model.

A literature review identifies the generic project life cycle stages and the role of SE in designing complex systems. Analysis of Benchmark's project life cycle model and the existing engineering processes shows that many SE elements already exist within the company. However, a new SE program is needed to integrate and make these elements more explicit. Interviews with project managers and a survey confirm this need and indicate the importance of good communication, collaboration, traceability, thorough documentation, requirements management, and the responsibility of SE for ensuring quality.

The thesis concludes by presenting a vision for SE at Benchmark, an updated view of Benchmark's project life cycle model focusing on what program tasks belong to SE, and a formalized SE process to supplement the existing EE, ME, and SW discipline processes. Finally, it also briefly introduces the necessary next steps for implementing the new SE program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

A3AO	A3 Architecture Overview
BOM	Bill-of-Materials
CAD	computer-aided design
CCB	Change Control Board
CMMI	Capability Maturity Model Integration
CTQ	Critical-to-Quality
DDP	Design and Development Plan
DFM	Design for Manufacturing
DFMEA	Design Failure Modes and Effects Analysis
DFT	Design for Testing
DHF	Design History File
DMR	Device Master Record
DMS	Document Management System
DVP	Design Verification Plan
DVT	Design Verification Testing
EE	Electrical Engineering
EMC	Electromagnetic Compatibility
ERS	Electronics Requirements Specification
ESD	Electrostatic Discharge
EVA	Earned Value Analysis
EVP	Electronics Verification Plan
FAI	First Article Inspection
FDA	Food and Drug Association
FMEA	Failure Modes and Effects Analysis
FPGA	Field Programmable Gate Array
A3AO	A3 Architecture Overview
FTE	Full-Time Equivalent
HoQ	House of Quality
IDME	Industrial Design and Mechanical Engineering
IMP	Integrated Master Plan
IPD	Integrated Product Development
EVP	Electronics Verification Plan
IRS	Interface Requirements Specification
IPT	Integrated Product Team
IVVQ	Integration, Verification, Validation, and Qualification Plan
LMS	Learning Management System
MBSE	Model-Based Systems Engineering
ME	Mechanical Engineering
NMA	Need-Means Analysis
NPI	New Product Introduction
PCB	Printed Circuit Board
PCD	Product Concept Description
PDD	Product Design Description
PM	Program Manager
PO	Purchase Order
PR	Problem Report
PRS	Product Requirements Specification
PTP	Product Test Plan
QFD	Quality Function Deployment
QMS	Quality Management System
R&D	Research and development
RCA	Root-Cause Analysis
RMP	Risk Management Plan
SE	Systems Engineering
SW	Software
TPD	Technical Product Documentation
TPM	Technical Performance Measure
TRIZ	Theory of Inventive Problem Solving
TRL	Technological readiness level
V&V	Verification and validation
VOC	Voice-of-Customer
VTP	Verification Test Procedure
WBS	Work Breakdown Structure
V&V	Verification and validation

EXECUTIVE SUMMARY

This thesis focuses on exploring the benefits of SE for Benchmark Electronics, a worldwide provider of product design services, engineering services, and advanced manufacturing services. The objective is to design a comprehensive SE program for Benchmark Almelo that fits within the company's existing workflow.

SE is a multidisciplinary approach to designing and managing complex systems that incorporates various engineering, management, and organizational processes. It is becoming increasingly important for companies in the high-tech sector, like Benchmark, due to the growing complexity of modern technologies and the need to bring innovative products to market quickly and efficiently.

At Benchmark, no formalized SE process is currently in place and because of this, it is often unclear where the responsibilities of the systems engineer begin and end. This leads to arbitrariness of the SE role, which hurts continuity across projects, knowledge retention, and overall quality. On that account, there is a need to concretize a SE approach and way-of-working for the company.

The thesis addresses the following main research question: how can SE be effectively implemented in new product development at Benchmark? To answer this question, the thesis addresses several sub-questions related to SE, *good* SE, Benchmark's approach to new product development, areas for improvement, and how SE can improve these areas. The thesis presents a SE process that is compatible with Benchmark's existing processes and adapted to Benchmark's project life cycle model. It also briefly introduces the necessary next steps for its implementation.

A theoretical framework (Part I) was established first to lay the groundwork for answering these questions. The system design process and the SE role were explored via literature review. Based on a variety of industry and academic sources, the generic project life cycle stages were identified as follows: *Concept*, *Development*, *Production*, *Utilization & Support*, *Retirement & End-of-Life*. During a project, the product or system under design progresses through these stages as it increases in maturity.

By carefully following these phases, project teams can ensure the successful development of a new or modified product or system. The systems engineer has an important role in guiding this. By researching SE, it was concluded that SE is not a traditional engineering discipline like mechanical or electrical engineering. Instead, SE is not limited to a single discipline and concerns itself also with technical management and business aspects. The goal of SE is to provide a quality product that meets user needs, by ensuring the right product is built in the right way.

Analysis (Part II) gave insight in what internal processes define Benchmark's design process. As most important were identified Benchmark's Engineering Design Control Methodology, and the existing Electrical Engineering (EE), Mechanical Engineering (ME), and Software (SW) discipline processes.

Benchmark's Engineering Design Control Methodology is an in-house designed project life cycle model consisting of seven phases. These are *1. Technology Development*, with the aim to demonstrate basic product, production process, and design feasibility, *2. Concept Development*, in which the technical concept is developed, *3. Design*, an iterative phase where the product is defined in detail, *4. Design Verification*, where the product is tested on its specifications, *5. Pre-Production*, where a pilot device is built to verify the manufacturing approach, *6. Qualification*, where the final qualification units are verified and validated, and finally

7. Production, in which the product is manufactured and lives on under strict change-control. It was important to become familiar with this model, since the new SE process must integrate with it.

Next, the EE, ME, and SW engineering processes were analysed. For each discipline, these cover the various activities and deliverables needed for successful design, verification and validation, and acceptance by the customer. They roughly follow a V-Model, with Waterfall and Spiral structures as intermediaries. Another important finding is that these processes remain quite general on how exactly to perform the activities they describe, but instead they refer to separate templates for more rigid guidance.

From this analysis it became clear that many SE elements already exist within the existing processes and procedures. The goal of the new SE program will be to integrate these elements and to make them more explicit. As part of this, a SE process shall be designed in the image of the existing discipline processes. It must link these processes together and describe for the SE function what to do, and when.

Additionally, interviews and surveys were conducted to supplement the research into the above processes and standards and to assess how well Benchmark performs SE and to find out what may be improved. Program Managers (PMs) from a variety of projects were selected to give input on a large list of topics, following a semi-structured interview procedure.

The results showed that with respect to SE, the PMs hold the following in high regard: the importance of good communication and collaboration, the need for a concrete SE role in a project with responsibility for requirements and design, the need for traceability and thorough documentation, tooling for tracking of requirements, the value of collaboration and top-down management by the SE, and the role of the SE in ensuring quality in the project. The PMs also all agreed that the SE should have a broad understanding of technology and that they should be very experienced.

Synthesis (Part III) was then carried out, combining the analysis, interviews, and survey to design a new SE program for Benchmark. First a vision was formed, comprising the deliverable of the thesis. Next, a revised Engineering Design Control Methodology was made, including the tasks and responsibilities of the SE in Benchmark's life cycle model. This is the second deliverable of the thesis. Lastly, in a similar way to the existing discipline processes, a SE process is presented. This process is the final deliverable of the thesis.

The SE vision for Benchmark was drafted together with an internal focus group consisting of system architects, competence leadership, and lead engineers. The vision can be summed up as follows:

"Systems engineers should be responsible for a design on system level, ensure proper communication and overview of the technical solution and trade-offs, monitor risks and changes to prevent unforeseen errors, and ensure proper testing and verification of the solution."

Next, to supplement Benchmark's life cycle model, the SE program tasks within the system life cycle were identified. Forming the responsibilities of the SE function within a program, the following tasks were defined:

1. *Perform Solution Synthesis*
2. *Perform Requirements and Technical Performance Measure Analysis*
3. *Perform Functional Analysis and Allocation*
4. *Create a Concept Design Description*
5. *Create and maintain a System Architecture Overview*
6. *Support Verification and Validation Plan Creation*

7. *Support Synthesis, Analysis, Evaluation*
8. *Support Risk Management activities*
9. *Plan, coordinate, and conduct formal Design Review meetings*
10. *Lead integration and monitor testing activities*
11. *Ensure proper documentation*

Apart from this generalized look on the SE tasks within a program, a formalized SE process was created in order to supplement the existing discipline processes. Like the EE, ME, and SW processes, it follows an activity flow-chart, supplemented with an extensive set of deliverables. The following activities have been defined in the process:

Kick-Off
Solution Synthesis
Requirements and Functional Analysis
Concept Development
Requirements Specification
Detailed Design
Design Review
Design Verification and Validation
TPD check
Design Release and start of Life Cycle Management

The defined vision and identified SE program tasks were discussed extensively in the focus group discussions and the formalized SE process underwent a call-for-review according to Benchmark's review procedure. These measures ensure that the new process has inherent support from the most important stakeholders; an important step in the change management that is to come.

Finally, while a pilot and further change management was in the end deemed out of scope for this thesis, the thesis presents the four steps for successful implementation: *stakeholder and change management*, *testing and iteration via a pilot*, *key trainings for implementation*, and the necessity for *monitoring and continuous improvement*.

In conclusion, the thesis presents a well-defined vision for SE within Benchmark, which is supported by a focus group of select internal stakeholders. Furthermore, the identified SE program tasks and formally defined SE process form the basis of the envisioned implementation, based on the academic state-of-the-art, all the while taking advantage of the SE that already exists. These, together with an outline of the to-be-taken next steps, allow Benchmark to continue in the future with the implementation of SE as described in this thesis.

1. GENERAL INTRODUCTION

1.1. Benchmark Electronics

Benchmark Electronics is a worldwide provider of product design services, engineering services, and advanced manufacturing services. It has been providing these services to original equipment manufacturers since 1979 and serves the aerospace and defence, medical technologies, complex industrial, semiconductor capital equipment, telecommunications, and advanced computing industries.

The company's core strength lies providing concept-to-production solutions. Benchmark's global manufacturing presence enables accelerated time-to-market, especially for complex products with lower volume and higher mix in regulated markets such as aerospace or medial with higher reliability requirements. This, coupled with in-house engineering capabilities, including product design, allows Benchmark to become an integral part of its customer's business.

The branch in Almelo is mainly active in the medical technologies, complex industrial, and semiconductor industries, with a focus on R&D and new product development. What makes the Almelo branch special is the integration of manufacturing and R&D in one building. Quality is valued highly and the company hosts a pleasant, collaborative environment. The working atmosphere at Benchmark is informal and collegial, and there is a definite Twente-like character. People enjoy a challenge. Finally, the organisation in Almelo is not very hierarchical, leading to a culture that fosters bottom-up innovation.

One of the main challenges faced by Benchmark is the ability to maintain technological and manufacturing process expertise. Since Benchmark's markets are characterized by rapidly changing technology and new process development, it is necessary to continually evaluate the advantages and feasibility of new technologies and processes to meet the customers' changing needs. To achieve this, at least in new product development, a well-structured SE process is necessary.

Some examples of products designed by Benchmark Almelo are shown below in Table 1. Each inhibit complex electronics and software, some form of connectivity or interfacing with the use environment, user-interface design, and industrial design. Next to Almelo, Benchmark Electronics has twenty-two other branches, across the Americas, Europe and in Asia. Some of the products below are designed in Almelo and produced in another branch.

From here on, the name Benchmark shall refer exclusively to the branch in Almelo, unless otherwise stated.

Table 1. Examples of products designed by Benchmark Almelo

		
DNAnudge Lab-free COVID-19 Tester	**Bambi Medical Wireless Monitoring System**	**Abionic AbioScope Microfluidic Analyzer**
Lab-free COVID-19 testing in use by the NHS, for up to 4 people, on-the-spot, at the same time.	*Wearable device for neonatal vital signs monitoring of babies with a wireless monitor.*	*Point-of-care microfluidic analyser for a variety of tests for infectious and cardiovascular disease, immunology, critical care, and drug monitoring.*
		
Malvern Panalytical XRD Analyser	**Fluke Industrial Acoustic Imager**	**Setra Power Meter**
Lab-based X-ray diffraction analyser to analyse solid and liquid samples on physical properties such as phase composition, crystal structure, powder orientation, etc.	*Handheld device using a microphone array to visualize sound (e.g. coming from small leaks in compressed air, gas, and vacuum systems).*	*Industrial power meter with high connectivity over serial EIA-485 or Ethernet, multi-load monitoring, data logging, web interface, field configurable, enabling safe and accurate measurement of both low and high amperage services.*

1.2. Research objectives

SE is a multidisciplinary approach to designing and managing complex systems that incorporates various engineering, management, and organizational processes. It is becoming increasingly important for companies in the high tech sector like Benchmark, due to the growing complexity of modern technologies and the need to bring innovative products to market quickly and efficiently.

Because of the above challenges, there is a need for Benchmark to concretize its SE approach and way-of-working. While a draft SE process was created back in 2014, it was never completed and implemented [1]. Thus, currently a formalized process is not in place. Because of this, it is often unclear where the responsibilities of the Systems Engineer begin and end. The ambiguity of the SE role hurts continuity across projects, knowledge retention, and overall quality.

Although it does use the deprecated 2014 SE Process as input, this thesis aims to re-evaluate Benchmark's SE approach from the ground up. The objective is to explore the benefits of SE for Benchmark and to design a comprehensive SE program that fits within the company's existing workflow. Part of this is to create a vision, analyse the current situation, and to design a strategy for implementation.

Learnings discussed by S. Jackson [2] and De Landtsheer et al. [3] have informed this general approach and led to the creation of the research questions. The main research question is: *How can Systems Engineering be effectively implemented in new product development at Benchmark?*

To answer this question, the thesis will address the following sub-questions:

A. What is Systems Engineering?
B. What is *good* Systems Engineering?

C. What is Benchmark’s approach to new product development?
D. What areas can be identified for improvement?

E. How can Systems Engineering improve these areas?
F. What should the Benchmark Systems Engineering approach look like?
G. How should this Systems Engineering program be implemented within Benchmark's existing workflow?

1.3. Thesis contents

The thesis is divided into three sections. In Part I, a theoretical framework is established to provide a comprehensive understanding of the subject matter and address questions A and B. This is achieved through a literature review in Chapters 2 and 3 that focuses on the design and development process and familiarizes the reader with SE.

Part II focuses on analysing Benchmark's current processes and identifying areas for improvement. This is achieved through a study of Benchmark's current workflows and procedures in Chapter 4 and interviews with key stakeholders in Chapter 5. The findings from this section will answer questions C and D.

Part III integrates theory with practice in Chapter 6 by synthesizing a concrete vision for SE and identifying SE program tasks. The vision and identified program tasks are then combined into a new SE approach that is presented in Chapter 7. Finally, although the pilot testing of this approach is outside the scope of the thesis, Chapter 8 provides guidelines for designing and evaluating a pilot, as well as steps for implementation and ongoing improvement.

PART I — Theoretical framework

Chapter 2 introduces the system life cycle stages, which form the basis of the system design process, each with its own specific goals, requirements, and challenges. These are important to understand, in order to see how SE plays a role in the design process.

Chapter 3 delves deeper into what is a system and lays out the various academic and industry definitions of SE. There are many available, so to establish one definition for the rest of the thesis, common threads are identified. Next, the various SE process models are discussed and what types of SE implementation may exist within a design team. It also forms the basis of how SE capability can be assessed by introducing a SE capability model.

2. THE SYSTEM LIFE CYCLE STAGES

To understand SE, it is first necessary to understand the project or system life cycle. Each system has a life cycle, even if it is not formally defined. The life cycle of a system can be defined as the set of all maturity stages through which the system passes, sequentially from cradle to grave. This chapter gives an introduction to understand the system life cycle.

A system is a set of interconnected parts that work together to achieve a specific goal or function [4] [5]. It can be physical, biological, or conceptual, and can range in size from small and simple to large and complex. For the purpose of this explanation, it will be regarded as a product that may involve software, hardware, mechanical, or electrical elements. Examples include a computer system, an HVAC system, or a manufacturing process.

For these kinds of engineering systems, many different life cycle models are available. Among the more commonly used ones are the System Life Cycle Processes by ISE/IEC/IEEE [6], NASA Program/project Life Cycle [7], and the US Department of Defence Acquisition Process Phases [8], but they generally all follow the same structure: from the design and realization of the system in the *concept, development*, and *production* phases, all the way through the system's *utilization/support*, and ultimately to *retirement* (Figure 1). Understanding these stages and the activities therein is crucial in understanding SE and where it plays a part.

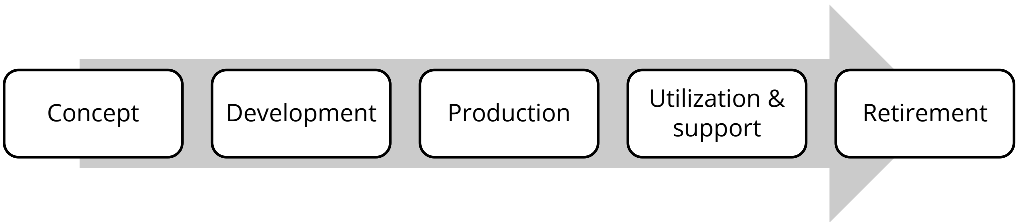
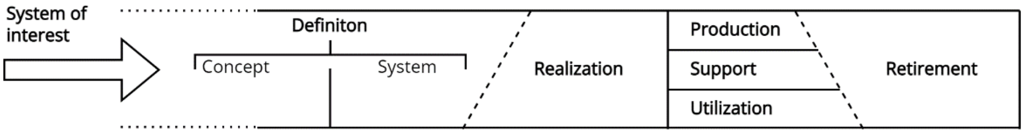


Figure 1. Generic system life cycle

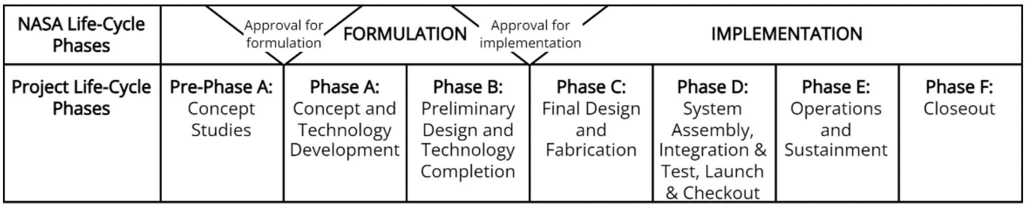
While the life cycles stages themselves can be defined differently per organisation, the overarching idea is the same: as stated excellently in the INCOSE Systems Engineering Handbook, "the needs of each subsequent life cycle stage must be considered during the earlier stages … in order to make the appropriate trades and decisions to accommodate the needs of later stages in an affordable and effective manner" [9].

Figure 2 presents a set of relevant system life cycle models, as synthesised from INCOSE [6], ISO/IEC/IEEE [9], NASA [7], Blanchard and Fabrycky [4], US Department of defence [8], and the SE Book of Knowledge [10]. These sources are important to highlight because they are recognized as authorities in the field of SE and provide comprehensive frameworks and guidelines for SE. Additionally, they offer diverse perspectives and examples that can be used to inform the thesis research and provide a solid theoretical foundation for the study.

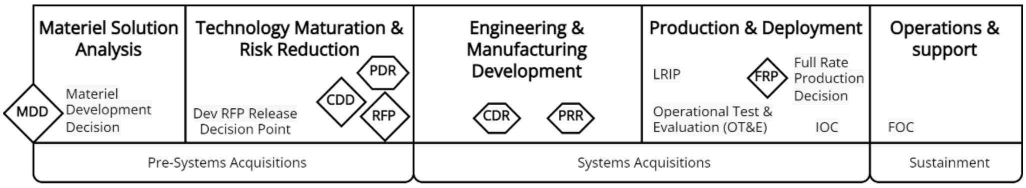
A. Generic life cycle model published in ISO 15288:2015 (figure reproduced from [6])



B. Generic life cycle model presented by SEBoK (figure reproduced from [11])



C. NASA Program/project Life Cycle (figure reproduced from [7])



D. Acquisition Process Phases as presented by the US Department of Defence (figure reproduced from [8])

Figure 2. Comparison of life cycle models

The next sections give an overview of the generic life cycle model phases distilled from these sources and the activities therein. They combine the learnings from the above sources into one a generic life cycle model description, to serve as the basis for understanding the system life cycle.

Benchmark also has its own system life cycle model, which is introduced in Chapter 4.

2.1. Concept phase

Generally, the concept phase starts with a recognition of a need for a new or modified system or product. In this phase, the available technology is explored to assess the TRL. Then, through the process of requirements elicitation, initial critical requirements for the new system are identified and defined.

As part of this process studies should be performed into context, opportunities, and stakeholder needs. It is important to consider these factors early in the life cycle so critical requirements are uncovered in time to ensure proper coverage in later phases. If work is done properly in this early phase, it is possible to avoid rework and recalls later.

Next to requirements elicitation, the concept phase starts with an exploration into technologies which may integrate into the solution. Tools such as TRIZ [12], Systematic Inventive Thinking [13], morphological charting, financial forecasting, Pugh comparison [14] [15] [16], FunKey Architecting [17] [18], Quality Function Deployment [19] [20], and many others may be used in this phase to come up with a solution to fill the need.

With a high-level early concept in mind, studies can then be performed into risks and challenges early, to avoid later issues. Risk reduction activities in this phase such as go/no-go assessment, expert reviews, and DFMEA are paramount.

Not taking enough time in this phase may lead to poor estimations and projections. Another pitfall may be that when this phase is rushed, there is a poor understanding of the technical solution space and therefore poor understanding of technical alternatives when performing trade-off studies between technologies.

The end of the concept phase comprises concept selection. For this, in-depth studies are performed on candidate concepts to provide a substantiated justification of the final system concept (or combination thereof) to be selected for development. Choices should be backed by architectural models, computer models, experiments, prototypes, etc. and issues related to integration must not be ignored since they can be discriminators in concept selection.

Thus, the key endpoint of the concept phase is a substantiated confidence that the business case is sound and the proposed solution is achievable. The project team should have a clear view of customer wishes and general requirements, program plan and budget, verification of the technology readiness level, threat analysis, and also already an idea of the bill-of-materials and related manufacturing processes and constraints.

In short, common elements of the concept phase are research to 1) define the problem space, 2) characterize solution space through technology exploration, 3) identify business needs, and 4) without doing any design work, to estimate budget and timeline.

2.2. Development phase

The development stage is where the product is developed. The selected concepts are taken as input from the previous stage and are elaborated in detail down to the lowest level. This phase is iterative and focused on detailed design, prototyping, and documentation. The goal is to produce the solution that meets the stakeholder requirements. For this it is vital to continue with user involvement through in-process review, approval, and control.

Multiple rounds of prototyping typically occur in this stage, with each round focusing on increasing comprehensiveness of the design requirements and more thorough testing. This, alongside simulations and modelling, aims to ensure substantiated coverage of stakeholder requirements.

During the development stage, the design of the system progresses through increasing levels of depth and detail, with multiple formal reviews to ensure the design solution is unambiguously defined and meets the design requirements.

During this phase much design data is created, such as design descriptions, drawings, CAD models, testing and validation procedures, risk management procedures, bill-of-materials, supply-chain planning, and manufacturing planning and process descriptions.

The output of this stage is a design freeze where the design, specification, and documentation are suitable for production building and testing. TDP shall be sufficiently and formally verified before entering the production stage. Insufficient control or premature succession to the next stage can lead to losses, recalls, and ultimately project failure.

2.3. Production phase

In the production stage the system is built or manufactured. The project team should support manufacturing and solve any issues as they may arise. Design modifications may still be necessary in this stage, to lower production costs, overcome issues, or enhance system performance. However, strict change control in this phase is necessary as any modifications may influence system requirements and require re-verification and validation.

2.4. Utilization and Support phase

In this stage the system is in use and activities are related to sustaining the operation of the product. Modifications may be needed to resolve support, compatibility, or security problems, reduce operational costs, or extend the life of the system. In any case, strict change control must be in place to avoid loss of the system's capability to satisfy requirements while under operation.

2.5. Retirement and End-of-Life phase

As part of this stage, the system and its components deprecated and no longer in use. It is important to consider this stage in the concept and development phases, in order to facilitate sustainable end-of-life. Remaining activities in this stage ensure end-of-life requirements are met.

2.6. Conclusion

In conclusion, the development of a new or modified system or product involves several distinct phases, each with its own specific goals, requirements, and challenges. The development process of a system is important to understand, in order to understand how SE plays a role in it. There are many life cycle models available, but a common thread can be identified.

The *concept phase* is critical as it sets the foundation for the entire project, including identifying stakeholder needs, exploring available technologies, and developing a high-level concept. The *development phase* focuses on detailed design, prototyping, and documentation, with the goal of producing a solution that meets stakeholder requirements. The *production phase* involves building or manufacturing the system, with strict change control necessary to avoid adverse impact on system meeting its requirements. The *utilization and support phase* concerns sustaining the operation of the product. Finally, activities in the *retirement and end-of-life phase* are crucial for ensuring sustainable end-of-life, with activities aimed at meeting end-of-life requirements.

It is important to note that the processes within these phases are iterative in nature, and some processes can and do span multiple phases. For instance, requirements definition spans the concept and development stages, as requirements are substantiated to an increasing level of detail.

All in all, by following these phases carefully and thoroughly, project teams can ensure the successful development and implementation of a new or modified system or product. The SE has a strong role in guiding the project through these phases. The next chapter will discuss SE in detail.

3. SYSTEMS ENGINEERING

SE is about creating successful systems by properly 'guiding' the system through its life cycle. It is a difficult to define profession, but its goal is to achieve *balance*, *coherence*, and *integration* between system specifications and solutions, and make sure that subsystems properly work together to serve a singular goal. There are three pillars that SE is based on: processes, tools, and system thinking.

This chapter forms the basis to understand SE: what is a system, and what is SE? The existing industry-standard definitions are discussed and although not one is definitive, commonalities are identified. Next, the various SE process models are discussed and what types of SE implementation may exist within a design team. To help define what best-practice process areas belong to SE, an industry-standard SE capability model is introduced. Lastly, MBSE is briefly introduced, showing a glimpse of the future of complex SE.

3.1. Definitions

3.1.1. What is a system

To understand SE, we must first understand what is a 'system'. A system is generally considered to be a collection of interconnected elements or components that work together to achieve a common goal or purpose [4]. In other words: the sum is greater than the parts. This is the case for many contemporary engineering systems. For instance, an optical spectrometer combines a set of mirrors with a light detector and a specialized processing unit into a device that can separate wavelengths and measure light intensities.

There are many different official definitions, each highlighting different aspects of what constitutes a system. For example, ANSI/EIA-632-1999 defines a system as "an aggregation of end products and enabling products to achieve a given purpose," while ISO/IEC 15288:2008 defines it as "a combination of interacting elements organized to achieve one or more stated purposes" [21] [6]. The INCOSE Systems Engineering Handbook defines a system as "homogeneous entity that exhibits predefined behaviour in the real world and is composed of heterogeneous parts that do not individually exhibit that behaviour and an integrated configuration of components and/or subsystems" [9].

Concluding, the various definitions of a system all stress that a system is a collection of interconnected components or 'subsystems' that work together to achieve a common goal or purpose, and that the interactions between those components are what give the system its unique properties and abilities.

3.1.2. What is Systems Engineering

SE concerns the process of creating complex systems. It emerged as a transdisciplinary approach to manage complex and ever-changing technical projects. It focuses on balancing stakeholder needs and success criteria, starting early, but encompassing the entirety of the development cycle [22]. In the end, it allows for effective management of system verification and validation (see Box 1).

Box 1. Verification and Validation

> The difference between verification and validation is important to understand. As defined and taken verbatim from IEEE: "validation is the assurance that the system meets the needs of the customer and other identified stakeholders" [20]. This means validation is often done with or by the customer or end user, as it is the final check to show the designed system meets its goal.
>
> In contrast, again per IEEE: "verification is the evaluation of whether or not a system complies with a regulation, requirement, specification, or imposed condition" [20]. This is often an internal process, to ensure that the designed system or subsystem meets its requirements. It can be a proxy for validation, but only if the requirements set is properly defined. Therefore, in practice, both verification and validation occur.
>
> In summary, validation ensures that one is working the right problem, whereas verification ensures that one has solved the problem right [57].

To this day there is no established universally accepted definition for SE. How SE is applied in practice is largely based on the background and experience of the individual or organisation in question. Hendrik W. Bode, widely regarded as one of the pioneers of modern SE, stated in 1967 that "Systems Engineering is an amorphous, slippery subject that does not lend itself to such formal, didactic treatment [as defining it]" [23]. This holds true today. Nevertheless, several bodies of authority on the matter have made good attempts:

1. **INCOSE** defines it on their website as 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" [22].
2. In the **INCOSE SE Handbook**, the definition is more extensive: it is defined as "an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs" [9, p. 265].
3. **NASA** defines it as "a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system" [24].
4. **ISO, IEC, and IEEE** define SE as an "interdisciplinary approach governing the total technical and managerial effort required to transform a set of customer needs, expectations, and constraints into a solution and to support that solution throughout its life" [25].

These definitions share a number of common denominators. First, each definition explicitly states the inter- or transdisciplinary[1] aspect of it. Because SE encompasses the design of engineered systems as a whole, it inherently combines all participating engineering disciplines. The systems engineer must take into account considerations and constraints from

[1] while in some contexts the prefixes 'trans-' and 'inter-' are opposites, here the meaning is the same.

all disciplines, be it mechanical, electrical, software, test, production, but also sales and marketing and more.

Secondly, the system life cycle aspect is included in all definitions. SE not only concerns the design, synthesis, and realization of systems, but encompasses a broader view including use and end-of-life. This is important, as engineering trade-offs and business decisions often differ in outcome when applying a life cycle-centric view versus a limited view on only the design and development of the system.

Third is the notion that SE is a structured, scientific approach, driven by principles and methodology. Indeed, over the years, the field of SE has evolved to encompass a variety of methodologies, processes, and tools to develop complex systems while simultaneously aiming to improve clarity and communication within project teams. Among these are architecture frameworks, documentation standards, specialized SE software, and even system-modelling languages (see Section 3.5).

Finally, SE encompasses not only technical processes, but also managerial and business needs within a project. After all, the goal of SE is to provide a quality product that meets user needs. The goal of SE is to evaluate the stakeholders' goals early on in the development process, and to define these needs and subsequent required functionality and product requirements accordingly.

Now, it is clear that SE is not a traditional engineering discipline like mechanical or electrical engineering. Its implementation and degree of rigidness vary wildly between organizations and individuals. All in all, truly successful execution requires a well-planned and disciplined approach [2] [4] [26] [27].

Benchmark does not have an official vision on SE, but the above discussed definitions and identified commonalities will be used to define a vision on SE for Benchmark. This vision is presented in Section 6.1 and is drafted based on plenary focus group meetings with system architects, competence leadership, and lead engineers. This chapter was used as a primer for the discussions in the focus group.

3.2. Systems Engineering process models

As discussed in Chapter 2, the development team follows various processes in each phase of the system life cycle, in order to define complex systems from fuzzy to detailed. Next to modelling the system life cycle itself, it is possible to model how the processes therein should occur. For instance, as part of the *detailed design* life cycle phase, there is an ongoing prototyping process. The uncertainty and risk associated with system design, especially in the early project phases, make it necessary to follow a disciplined approach for the processes that occur as part of the system life cycle.

The most common process-models include the Waterfall Model, Spiral Model, and V-Model. In practice, a combination is often applied. Typically defined process models begin with the development or revamping of requirements, down to design, testing, and verification and validation.

3.2.1. Waterfall Model

The Waterfall model (Figure 3, [6] [9] [4]) originated in the world of software development as one of the first ever process models. It is linear and sequential, meaning each phase must be completed before the next phase can begin. The phases or activities do not overlap.

This one-way cascading progression means that when changes occur later in the project, there must be a fallback to a previous phase in the project. Furthermore, the Waterfall model does not facilitate for incomplete development stages. This combination makes the model largely unsuitable for complex projects with high risk or uncertainty.

However, it is extremely suitable for small sub-projects like the development of a simple test setup or parts of a larger software package. In these cases, the 'project' is clear and manageable, foregoing the need for a more complex approach like the V-Model.

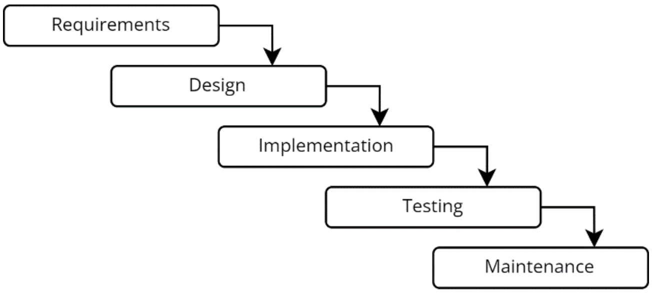


Figure 3. Waterfall model

3.2.2. Spiral Model

The Spiral Model (Figure 4, [6] [9] [4] [28]) combines the systematic, controlled phase progression of the Waterfall Model with the idea of iterative development. Four phases are executed sequentially for each cycle. Generally these consist of 1) planning the cycle and identifying its objective, 2) identifying risks, 3) development, and 4) review. However, models tend to differ based on the application. With each cycle of the spiral, the project is developed more in depth.

The Spiral Model is effective in managing projects with rigid budgets where risk evaluation is important, due to the frequent evaluations and reviews. However, while working according to this model it is important to identify clear goals and end-conditions, to avoid the spiral going on endlessly. Furthermore, it is plagued by its need for documentation at each of the many intermediate stages.

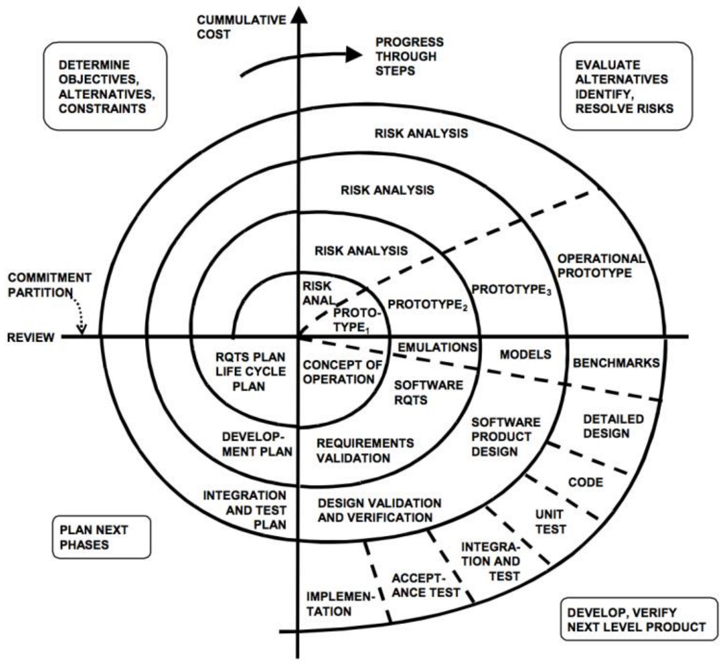


Figure 4. Spiral model
(figure reproduced from [28])

3.2.3. V-Model

The V-Model (Figure 5, [6] [9] [4] [10] [29]) visualizes the design and realization of a system on two diagonals. The left side of the 'V' represents the decomposition of requirements and creation of system specifications, from identification of need to technical concept, requirements, system architecture, down to the practical implementation of each subsystem. The right side of the 'V' represents the various testing and integration rounds that end in a verified and validated solution.

The V-Model emphasizes requirements-driven design and testing: requirements are elaborated from system level down to a monodisciplinary level, with traceability through all stages. All requirements must be traceable to one or more top-level system requirements and each requirement must be addressed by at least one acceptance test.

A downside to the model is that in principle it is not feasible to go backwards in the model. For instance, when a product is in the testing stage it is difficult to go back and change a functionality or operating principle, as much documentation will need to be updated and re-reviewed.

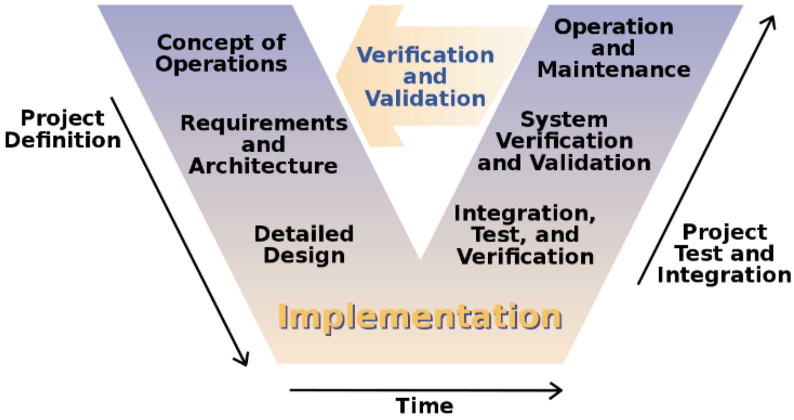


Figure 5. V-Model
(figure reproduced from [29])

3.3. Understanding Systems Engineering implementation

Many organizations implicitly practice SE (sometimes unknowingly) as part of their established development process. Especially modern practices in Integrated Product Development ('concurrent engineering' [30]) share many common threads with SE. Furthermore, widespread quality standards such as ISO 9000 inherently mandate the use of SE equivalent practices, even if a rigid SE way-of-working is not in place [31].

Due to this ambiguity, based on personal experience of speaking with system engineers at universities, symposia, and Benchmark, many questions arise when SE is discussed: What is SE? Is it a discipline, a process, an approach, a program phase? Is SE not simply concurrent engineering? What is the value of performing SE? What tools does SE use? Should SE be implemented the same way in all organisations?

The answer to these questions differs greatly per company and per person: for one individual, SE comprises simply the process of turning a complex problem into an objective statement of function, for others SE is more about management of stakeholders and their wishes, and in some organisations SE is regarded simply as the process of tackling technical problems in a structured manner and ensuring documented verification and validation.

This section aims to present some perspectives on how to see and talk about SE, partially answering the questions above. The mail goal is to serve as a 'primer' for internal discussions at Benchmark.

3.3.1. Sheard's three types of Systems Engineering implementation

To tackle SE's ambiguity Dr. Sarah Sheard defined three SE implementation types [30]. According to Sheard, while the total profession of SE encompasses all as a whole, it can be divided into three types of implementation each for a different depth and complexity of a project: *discovery*, *program management*, and *approach*. Despite the publication being almost a quarter of a century old, these implementation types are still useful today to understand the different views people have on SE.

Sheard argues the following: *Discovery SE* is the specialist-type implementation, focussed on concept exploration and verifying what problem is to be solved. *Program SE* is a more generalist approach, focussed on the technical side of program management, maintaining focus on customer need, and providing a cost-effective solution on time. Finally, *Approach SE* is even more general. Sheard describes Approach quite simply as "what every engineer should do" [30, p. 4]. Approach is driven by processes and the scientific method: not jumping to a solution, and maintaining focus on what the customer really wants.

These three types of SE also each prevail under different project circumstances. Figure 6 shows how the three implementation types overlap. In the figure, the horizontal axis represents the product life cycle, indicating where in the product life cycle the SE type is most relevant. The vertical axis indicates the complexity level in which the project resides.

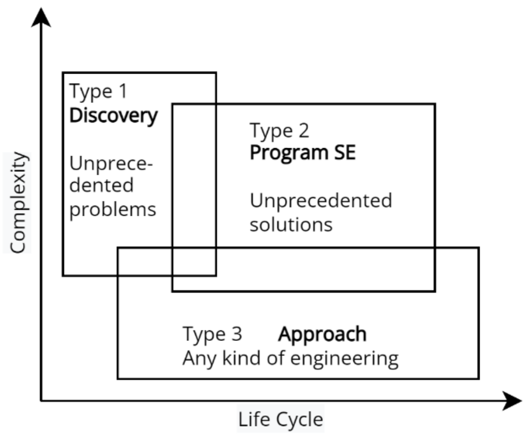


Figure 6. Three types of Systems Engineering implementation (reprinted from [30] with permission)

Naturally, Approach SE spans almost the whole life cycle. However, when project complexity increases the other two types also increase in importance. In very complex projects Discovery SE techniques are necessary to crystallize the fuzzy front-end of the project before the other types can commence. Later in the life cycle, Program SE replaces Discovery in managing the complexity and stakeholder needs as the project carries on.

In conclusion, while these implementation levels were never officially adopted by INCOSE or other SE authorities, many questions can be answered when SE is regarded as more than one single thing and these proposed types serve as an excellent framework for discussion. On the other hand, one must not become too locked-in to this segregation of the profession: while it does help in understanding the different views of SE, there are many overlaps and the overarching goal remains generally the same.

3.3.2. Systems engineering as design and management

Another possible way to divide SE is based on the objectives of the activities performed. I propose a division based on 'design' activities and on 'management' activities.

The design part of SE is about understanding the system context, problem domain, and exploring what exactly is the right problem to solve, and with what technology. SE is applied to identify what problem to solve, verify that the problem can be solved, and to determine how to solve it. Key activities are more design-focused, such as system analysis and modelling, eliciting and formulating system requirements, assessing the TRL, and making trade-offs.

The management part of SE is prevalent during the whole of the life cycle until the utilization phase. SE is used to ensure stakeholder needs are met in a cost-effective and timely manner. Key activities are more management-focused, such performing mission analysis, managing communication between stakeholders, maintaining budget and planning, and monitoring risks.

Sheard's Approach Systems Engineering is explicitly excluded from this division. Engineering efforts are generally already always driven by existing process descriptions. Furthermore, the prevalence of quality standards such as the ISO 9000 family make this type difficult to distinguish from general good engineering practices, raising the question whether this type of implementation requires its own category at all.

3.3.3. Conclusion

Regardless of the position in the system life cycle, SE can and should be applied. There can be much discussion regarding what activities are and are not SE, and much is engrained already in the scientific method of good engineering practices. This section has aimed to provide a primer to serve as common ground for discussion.

SE's implementation may be categorized in a variety of different ways, but the goal of the SE is clear: identifying the *right product* to be built all the while ensuring the product is *built right*. The combination of these two is the essence of the profession: a top-down approach, inherently interdisciplinary due to its complexity, focussed on the whole product life cycle, ensuring systems are properly designed according to the proper requirements.

3.4. Systems Engineering Capability Maturity Model Integration

Despite the broad attribution of what SE exactly entails, certain process areas within the system life cycle can be identified as typical areas where the systems engineer plays a significant role. It is necessary to identify what process areas are relevant for SE, since this allows for assessing the SE capabilities of an organisation. If the SE process areas are properly represented, it is possible to state that 'good' SE is applied.

Elm et al. previously published a survey on SE effectiveness for the Software Engineering Institute [32]. This publication is important input for this thesis, as it lays out a tried and tested way of assessing SE capabilities. To design the survey that was conducted, Elm et al. created a list of SE process areas. This list (and survey, see Section 5.1.2) will be used in this thesis as a basis. It is shown in Table 2.

To define this subset list, Elm et al. referred to the Capability Maturity Model Integration for SE, Software, and Integrated Product Development defined by the Software Engineering Institute [33, pp. 79-484]. Therein researchers from Carnegie Mellon University have defined a list of process areas relevant for Integrated Product Development, based on ISO/IEC/IEEE 15288 [6]. This is a long list consisting of 614 practices needed to satisfy 179 goals organized into 25 process areas. The CMMI also lists 476 typical work products produced by these practices.

Elm et al. created a subset of these work products by determining for each work product if it has a relation to the system as a whole or not. The process was as follows: first, all work items were extracted from the CMMI. Then work items were selected for the subset if they result from or are a part of the above SE principles, and if they are significant.

With this set of process areas in mind, an assessment can be made to assess the SE Capability of a company. For this assessment, one must investigate to what extent these process areas are represented and to what quality the deliverables are executed. Benchmark's SE capability is assessed in Chapter 5.

Table 2. Process areas and associated work products relevant for SE

Project planning ▪ Cost and schedule baselines ▪ Integrated master plan ▪ Integrated master schedule ▪ Integrated product teams ▪ Technical approach ▪ SE master plan ▪ SE master schedule ▪ SE processes ▪ Work breakdown structure	**Requirements engineering and management** ▪ Customer requirements specification ▪ System requirements specification ▪ Use cases ▪ Requirements acceptance criteria ▪ Requirements allocations ▪ Requirements approval process ▪ Requirements impact assessments ▪ Requirements management system
Project monitoring and control ▪ Earned Value Analysis (EVA) ▪ Peer review plan ▪ Review of action items ▪ Review of issues ▪ Review process ▪ Review of selection criteria ▪ SE tracking records	**Project integration** ▪ Product integration process **Test and verification** ▪ Verification procedure ▪ Verification entry and exit criteria ▪ Verification criteria **Validation** ▪ Validation procedure ▪ Validation criteria
Risk management ▪ Risk list ▪ Risk mitigation plans ▪ Risk mitigation status	**Configuration management** ▪ Configuration baselines ▪ Configuration item list ▪ Baseline archives ▪ Baseline audit records ▪ Change control board
Architecting ▪ Concept of operations ▪ Product architecture ▪ Interface descriptions ▪ Interface control documents	**Trade-off studies** ▪ Alternate solutions ▪ Trade-off study reports

3.5. Model-based Systems Engineering

Model-Based SE (MBSE) is a method of SE that utilizes models as the primary means of representing and analysing a system. It is a relatively new approach that has gained significant attention in recent years due to its potential to improve the efficiency and effectiveness of SE [34]. It is different from 'engineering with models'. which has been a common practice in the engineering profession for decades. Instead, MBSE centres system design around an ever-evolving system model, which serves as the single source-of-truth over the course the project [35].

One of the key benefits of MBSE is that it allows for a more holistic and integrated view of a system, which can lead to better understanding and communication among stakeholders. By using a common model to represent the system, all stakeholders can have a shared understanding of the system and its requirements, which can help to reduce misunderstandings and errors. Additionally, MBSE can help to automate many of the tedious and error-prone tasks associated with traditional SE, such as requirements management and traceability.

Another benefit of MBSE is that it can facilitate early detection and identification of potential issues or conflicts within a system. By using models to analyse and simulate a system, it is possible to identify potential issues or conflicts before they become critical, which can help to save time and resources in the long run.

However, MBSE is not without its flaws. One of the main challenges associated with MBSE is the complexity of creating and maintaining models, which can be a significant undertaking. Additionally, MBSE requires a significant investment in tools and resources, which can be a barrier for some organizations.

The formal methods introduced by MBSE (e.g. prescribing modelling languages like SysML or UML) require significant support from the whole organization. This makes it a challenge to introduce, especially when 'regular' SE has not yet been formally embedded already in the organization, as is the case for Benchmark.

In conclusion, while MBSE has the potential to improve the efficiency and effectiveness of SE, it is not without its challenges and requires a significant investment and learning curve. Thus, for now, this thesis will focus only on the traditional SE and system thinking tools and techniques, in order to concretize SE within Benchmark and to give the Benchmark systems engineer a collection of tools to use and benefit from. Nevertheless, introducing MBSE could become a good future improvement, after Benchmark has gained more experience with SEs on its project teams.

3.6. Conclusion

The goal of this chapter was to provide a base understanding of SE and its goals. It has become clear that SE is not a traditional engineering discipline like mechanical or electrical engineering. Instead, it is interdisciplinary approach, combining the engineering disciplines. Furthermore, it also concerns itself with business aspects, all with the end goal of providing a quality product that meets user needs. Finally, it is not limited only to the design and realization phases, but instead covers the whole system life cycle.

This chapter also introduced the Waterfall, Spiral, and V- process models. These go one level deeper than the life cycle stages as discussed in Chapter 2 and model the processes within these life cycle stages. These process models are important to understand since they describe how development efforts transition through various levels of detail.

Next, serving mainly as a primer for discussion, this chapter presented some views on how SE can be implemented. It has become clear that SE activities can classified into some 'types' of implementation: some activities fall into a design category, some more in technical management, and some are simply what every engineer already does, because they follow predefined, scientifically-backed processes. Despite the various types of SE, the overall goal has become clear: to identify the *right product* to be built and to ensure the product is *built right*.

To answer the question of what is good SE, this chapter introduces the SE Capability Maturity Model Integration. A subset from a list of industry standard process areas and deliverables, determining which are most relevant for SE. Testing how well this list is represented at Benchmark allows for an assessment of its maturity in SE.

Finally, MBSE was briefly introduced. The interest surrounding MBSE has been rising over the past years, due to its potential to improve efficiency and effectiveness of SE. However, it is deemed too challenging to implement at this moment, and for now the thesis will focus only on the traditional SE and system thinking tools and techniques. Once Benchmark's experience with SE increases, MBSE could be considered as further innovation.

PART II — Analysis

Chapter 4 introduces the established procedures by Benchmark to ensure quality and compliance in the design and development process. Described in detail are the Engineering Design Control Methodology which describes Benchmark's life cycle phases, next the EE, ME, and SW Processes, and the Design Change Procedure. Lastly the chapter discusses the most important project deliverables necessary to understand Benchmark's system design approach.

Chapter 5 chapter aims to identify and analyze areas for improvement in the implementation of SE. Informal discussions, observations, and interviews with several PMs were used as input, and a detailed survey was designed for PMs who declined the interview. The interviews and survey determine the maturity level of the most important SE process areas and aim to elucidate improvement areas by collecting input from a range of projects.

4. EXISTING INTERNAL PROCESSES

Benchmark has several processes and procedures set in place in order to ensure quality and compliance during the design and development process. The internal procedures ensure compliance to a variety of external standards. These include general QMS requirements (ISO 9001), QMS requirements for Medical Devices (ISO 13485), QMS requirements for Aviation, Space, and Defence (AS9100), and QMS requirements for the FDA (21 CFR Part 820).

The following sections discuss the overarching Engineering Design Control Methodology, which describes Benchmark's life cycle phases, the three existing EE, ME, SW discipline processes, and the Design Change Procedure. Lastly, this chapter discusses the noteworthy project deliverables that are necessary to understand the discussed processes and procedures. Analysing these processes and deliverables allows us to understand how Benchmark approaches system design.

4.1. Engineering Design Control Methodology

The project life cycle stages as defined by Benchmark are documented in their Engineering Design Control Methodology. This procedure, known internally as the Benchmark 7-step process, describes the life cycle phases and activities that are to be performed for all design and development programs.

Figure 7 gives a graphical representation of the project phases as defined in the 7-step process and Table 3 presents an overview of general activities performed in each phase.

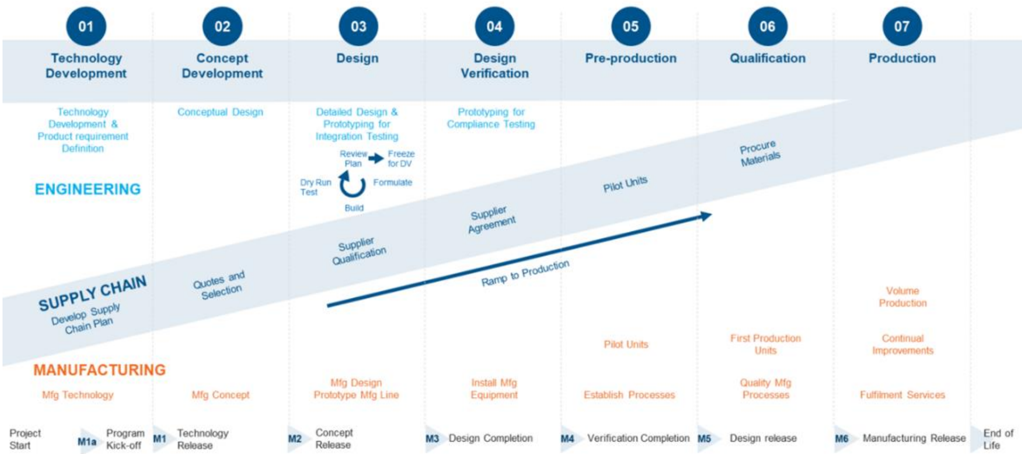


Figure 7. Diagram showing Benchmarks seven project phases (reproduced from Benchmark)

Table 3. Benchmark Engineering Design Control Methodology phases and activities

Phase	Name	Activities (generalized)
n/a	Kick-off/Project Proposal	Creation of project team Creation of DDP Define general design objectives
1	Technology Development	Define critical requirements Create PRS Proof-of-concept studies Program feasibility (TRL) and risk assessment Preliminary supply-chain assessment
2	Concept Development	Creation of DHF

		Define and update design input requirements Create and update risks and mitigation plan Preliminary DFMEA Usability assessment Create DVP Create conceptual designs
3	Design	Finalize PRS, MRS, ERS, SRS Finalize DFMEA Iterative cycle of prototyping, testing, and review at subsystem and integration level Design analysis (DFT, DFM, etc.) Supply chain analysis and supplier selection Update risk management and DHF as needed Update DVP and create verification traceability matrix Design freeze process for validation
4	Design Verification	Design verification testing and report Procure production tooling Procure components and verify conformity to specifications Final design reviews Update DHF as needed
5	Pre-Production	Support pilot device build Design validation and verification testing Update DHF
6	Qualification	Support qualification build Design validation and verification testing External agency testing, if applicable Design transfer, if applicable
7	Production	Product launch Volume production

4.2. Discipline processes

Next to the overarching 7-step design process, a step-by-step engineering procedure is defined for the EE, ME, and SW disciplines [36] [37] [38]. These procedures are not inherently linked to the 7-step process, but instead define the specific discipline processes in the overall development process.

The three discipline processes follow the same general outline:

- First, the PRS is translated into a requirements document for the discipline (xRS).
- From there, the processes move into a design phase, where detailed schematic diagrams are created, components are selected, and proofs-of-concept are built and simulations are performed.
- In a design review the design is checked by relevant team members and potentially the customer, to ensure that it meets all necessary requirements and constraints.
- Then, parts are verified and prototypes are built to be verified, first at discipline/component level and later at system level. If any requirements are not met, the design may need to be modified or the project requirements may be adjusted.
- All TPD created over the system life cycle is then collected and reviewed.
- Lastly, in a final review the design is released for manufacturing in the Design Readiness Review.

4.3. Design Change procedure

A properly defined change control procedure is necessary, to ensure compliance with 21 CFR Part 820 (FDA Quality System Regulation), ISO 13485, ISO 9001 and AS9100. The Design Change procedure concerns all design engineering teams for releasing and changing all controlled design and development documents. A CCB evaluates all proposed changes, to ensure there is no adverse impact on the conformity to requirements. Documents are released and revision controlled either in the Benchmark DMS, or a DMS prescribed by the customer.

4.4. Noteworthy project deliverables

The DDP, Design Input documents, Design Review reports, and the DHF are noteworthy project deliverables. The DDP is a document that outlines the design and development activities for a project, and must be approved by the PM, customer, and, in some cases, the lead engineer(s) and quality assurance team. The Design Input documents contain requirements and inputs from the customer, and are used to inform the design and development process. Design Review reports document the Design Review evaluations conducted during the design phase to ensure that the results meet the requirements. The DHF is a repository of all design and development documentation, including version history, and is used to demonstrate that the design was developed in accordance with the approved DDP.

4.5. Conclusion

This chapter presented a foundation to understand the current way-of-working at Benchmark. It has given insight into Benchmark's 7-step Engineering Design Control Methodology, the EE, ME, and SW discipline engineering procedures, the general Design Change procedure, and some of the most important project deliverables.

All of these are important input to Chapter 6 and the section about the discipline procedures is especially important input for Chapter 7. The new SE Process should integrate not only with the Engineering Design Control Methodology, but also connect with and 'talk to' the existing discipline processes.

The next chapter will, with the knowledge of these processes in mind, try to elucidate the shortcomings of the current way of working.

5. ASSESSING SYSTEMS ENGINEERING IMPROVEMENTS

The goal of this chapter is to identify and analyse improvement areas for the implementation of SE. Input for this consisted largely of informal discussions, observations, and experience from working at Benchmark. However, to substantiate the analysis, a more formal interview procedure was also applied. Several PMs were asked to participate in an interview and those who could not make it were asked to complete a detailed survey.

To gain the most information, PMs were invited from different projects of various size, risk, and budget. The goal of these interviews and the survey was to identify to what maturity level the process areas mentioned in Section 3.4 are executed. Next to SE maturity, the interviews aim to elucidate the complexity and success of the project in question. This is important to take into consideration since projects of different complexity and risk require a different level of SE implementation.

For the PMs that declined the interview, a detailed survey was designed. The goal of the survey was to record project complexity, success (financially, technically, timeliness), and the applied level of SE characterized per process area.

5.1. Methods

5.1.1. Interview

As stated, a selection of PMs were invited for an interview about (one of) their projects. The interviews were held in semi-structured form, instead of following a rigid set of questions. However, some general topics for discussion were pre-defined to roughly guide the interview and prevent it from going off-track, see Box 2.

Conducting the interviews in a semi-structured form allowed for flexibility in the line of questioning. This approach was chosen since it is beneficial for the assessment in a number of ways: first, it allows to explore the subject's experiences and opinions in greater depth, as it allows for follow-up questions on interesting or unexpected answers. This helps to elicit more detailed and nuanced responses from the subject. Secondly, flexibility in the line of questioning allows for adapting to the specific needs and characteristics of each PM, which improves the overall quality of the data collected. Lastly, the open-ended form of these interviews potentially reduces any social desirability bias.

The goal of the interviews was to understand what role is played by SE in the projects assessed, and possibly where SE mistakes were made—in essence a diagnostic interview. Next to specific insight in SE maturity of the implementation areas, the goal was to gain more general insight in the nature of the projects, management styles of the PMs, and other challenges in design engineering.

Box 2. Topics for semi-structured interview

General
- Project name, type? Customer? Market?
- Medical class? Avionics design assurance level?
- Start and end date?
- Team size?

Project complexity/challenge
- Technological readiness level; precedent for what is being done?
- Size of the development effort and team?
- Availability of resources and knowledge?
- Interoperability with other systems?
- Is there a separate SE function? And budget?
- Contract value and budget; changes due to?
- Customer satisfied in terms of cost?

Project planning
- Planning/duration; changes due to?
- Use of WBS that is well maintained with influence of SE and other stakeholders?
- Technical approach (HW, SW, ME dev plan.) tailored for project? Or generic?
- Schedule; is it event-driven? Waterfall Model or Spiral or V?
- Is the IMP consistent with WBS?
- Is the event-based schedule SMART? E.g. goals/criteria are measurable?
- Is there a masterplan that combines all?

Use of Integrated Product Teams
- Are IPT's used?
- Is there SE representation in each UPT?
- Does client and/or suppliers participate in IPT's?

Risk Management
- Is there a Risk Management process?
- How are risks documented, mitigated, and tracked?
- Is the Risk Management process integrated with decision-making? Cost and earned-value management? Scheduling?

Requirements Management
- Customer and product requirements? Regulatory, statutory, certification requirements?
- Operational concept? Use cases? Maintenance and support?
- Formal approval process?
- Requirements management system?
- Is customer satisfied with this project's performance with respect to satisfaction of requirements?

Trade-off studies and System Architecting
- Are trade-offs documented?
- Are interfaces described properly?
- High-level structure of the system? Is It maintained and stored? Multiple views (functional, modular, etc.)? Accessible for all?

Integration, verification, validation
- Entry- and exit criteria per phase?
- Integration process, plans, criteria, etc?
- Review process?

Configuration control
- Configuration baselines? Configuration management system?
- Change-control board?

Project Management and control
- Are cost and schedule baselines managed?
- Is earned-value analysis done? When is the EVA baseline updated? Is EVA linked to WBS and IMS?
- How are customer PRs handled?

Four separate PMs[2] participated in the interviews and two different PMs opted to fill in the Capability Assessment survey instead. Hierarchically, the PMs report directly to the director of program management and to the rest of the management team.

5.1.2. Survey

As previously discussed, PMs who declined the interview were requested to complete a comprehensive survey. The survey used was a modified version of the Software Engineering Institute's Systems Engineering Effectiveness Study [39], which was originally a part of the CMMI [33] [40]. The original survey was designed to evaluate companies on their SE implementation and maturity. Therefore, it was considered appropriate to use as a basis for the survey in this thesis.

The survey used in this thesis and the scoring procedure are documented in Appendix C. The survey consists of 107 questions in total, most of which are Likert-scale. It consists of the following categories (the first is general and the rest cover the SE process areas):

1. Project complexity and challenge
2. Project planning
3. Use of integrated product teams
4. Risk management
5. Requirements management
6. Trade-off studies and system architecting
7. System integration, verification, and validation
8. Configuration management and control
9. Project management and control

Each survey question is classified to contribute to one or more SE process areas. A score from one to four is attained for each question, and the unweighted average of a set of questions constitutes to the final scores per process area. Thus, a high score means good/'proper' representation of the process areas in the project and a low score means poor SE implementation.

5.2. Results

5.2.1. Interview

Due to confidentiality arrangements most detailed results are redacted.

A number of recurring themes can be identified in the interview results. The following were deemed to be the most relevant takeaways:

- **The importance of good communication and collaboration:** The interviews highlighted the importance of good communication and collaboration with clients and other parties involved in a project.
- **The role and responsibilities of team members:** The interviews discussed the different roles involved in a project, including the systems architect who is responsible for requirements and design and is closest to SE.
- **The importance of traceability and thorough documentation:** The interviews emphasized the need for traceability and maintaining thorough documentation in order to retain knowledge in long-running projects.
- **The use of tools for tracking requirements:** The interviews discussed the use of tools such as Word, Excel, and Atlassian for tracking requirements and the challenges of implementing automated testing.

[2] I have also spoken several times with the director of program management, as well as with other employees, however not specifically for these interviews.

- **The value of collaboration and a top-down approach to management:** The interviews discussed the importance of collaboration among team members and a top-down hands-on approach to management.
- **The role of Systems Engineering in ensuring project quality:** The interviews discussed the importance of traceability, thorough documentation, and the use of tools for tracking requirements, which are all key aspects of SE. They also mentioned in some cases these aspects may have been represented better, were there to be a dedicated systems engineer on the team.

5.2.2. Survey

The survey revealed that the process areas of *Integrated Product Teams*, *Risk Management*, *Trade-Off Studies*, *Requirements Management*, and *System Architecting* have the most potential for improvement.

5.3. Conclusion

The interviews conducted for this study provide valuable insights into the challenges and best practices in running projects at Benchmark. Common themes emerged related to the importance of good communication and collaboration, the need for a concrete SE role in a project with responsibility for requirements and design, the need for traceability and thorough documentation, tooling for tracking of requirements, the value of collaboration and top-down management by the SE, and the role of the SE in ensuring quality in the project. The interviewed PMs also all emphasized that the SE should have a broad understanding of technology and vast experience.

Due to the sample size of the survey it was not possible to conclude anything concrete from the survey results. However, combined with the conclusions from the interviews, it becomes clear that there is a need for a defined process to manage requirements and document trade-off studies. Introducing a dedicated SE role can improve the process area of *Integrated Product Teams* by providing technical leadership and increasing the effectiveness of communication and collaboration. Additionally, a proper SE process should tackle the challenges in *System Architecting*, *Risk Management*, and *Requirements Management*.

Figure 8 below shows how the key takeaways from the interviews and survey integrate into four concrete points for improvement.

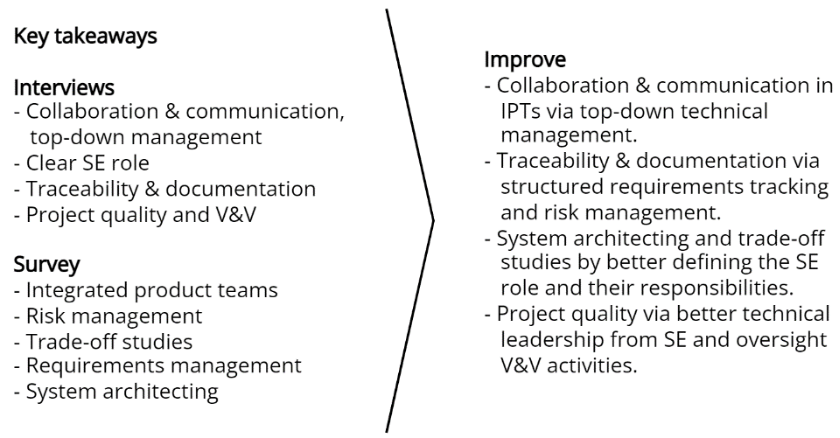


Figure 8. SE key improvement points

PART III — Synthesis

Chapter 6 defines the new SE vision and the responsibilities for the systems engineer. Input were industry best practices from Chapter 3, the analysis of Benchmark's existing processes from Chapter 4, and the capability assessment from Chapter 5. The chapter also discusses how to determine the level-of-detail for SE implementation per project, suggests how to structure SE within Benchmark's existing company structure, and presents an updated Design Control Methodology diagram.

Chapter 7 presents the formal new SE procedure for Benchmark, which replaces the deprecated draft SE Process from 2014. It ties together the existing EE, ME, and SW Processes, using input from Section 4.2. The chapter covers the new process diagram, deliverables, and important SE tools.

Chapter 8 presents a starting point for the change management approach to implement the new process. Successful implementation of the program requires testing, validation, training, and continuous improvement. This chapter presents the setup for a pilot project, improvement, and training and support.

6. STRATEGY MAKING

This chapter outlines the definition of the new SE process. The process is designed based on the insights and findings from the analysis of industry best practices (Chapter 3), an examination of Benchmark's existing processes (Chapter 4) and the capability assessment (Chapter 5).

The industry best practices and desk research provided a general understanding of the most common approaches to SE and helped shape the definition of the SE role and its capabilities. The existing processes and structure of Benchmark were crucial in forming the new SE process, as it needs to be compatible and work seamlessly with the current processes. By aligning the new process with Benchmark's existing way of working, it increases the chances of successful adoption.

The capability assessment showed the need for a clear and structured SE process, highlighting the importance of good communication and collaboration, traceability and documentation, and the necessity for the systems engineer to have a top-down approach to technical management and a broad understanding of technology. The assessment results provided the necessary guidance to tailor the new SE process to meet the specific needs of Benchmark.

This chapter opens with the definition of a vision, based on meetings with an internal focus group. Then, the SE theory is linked with Benchmark's Design Control Methodology, by first defining which program tasks therein should belong to the new SE function, and then presenting an updated Design Control Methodology diagram. Next, the level-of-detail for SE implementation is discussed, as this will vary for each project. Lastly, this section presents a suggestion on how to structure SE within Benchmark's existing company structure, by creating a new SE competence group.

6.1. Vision

A SE vision (Box 3) was established through various meetings with a focus group consisting of members of the MT, HW and SW architecting team, and PMs at Benchmark. To start, all participants were required to familiarize themselves with the theoretical framework by reading Chapters 2 and 3. This ensured that all members were on the same page regarding the academic state-of-the art surrounding SE theory.

In multiple sessions, the focus group reflected on the definitions as presented in Section 3.1 and 3.3. The group agreed that all views on SE were valid and confirmed the theories laid-out in Section 3.3, including Sheard's three views on implementation. They concluded that while some SE tasks may be more management-focused (risk management, forecasting, stakeholder management) or engineering-focused (integration, testing, verification), the SE role should integrate both. However, the technical engineering aspects always received more weight. The SE role was described as almost a hybrid between a lead-engineer and technical manager. Overall, the group found this holistic aspect of SE to be a valuable addition to the discussion.

Box 3. Systems Engineering vision

In the future, SE should be a part of Benchmark in such a way that its responsibilities and value are clear to all stakeholders, including all project engineers, PMs, and Benchmark's customers.

The SE role should radiate top-down multidisciplinary technical leadership through vast experience in completing complex multidisciplinary engineering projects. Systems engineers serve to ensure a proper balance between specification and execution.

"Systems engineers should be responsible for a design on system level, ensure proper communication and overview of the technical solution and trade-offs, monitor risks and changes to prevent unforeseen errors, and ensure proper testing and verification of the solution."

- **They are responsible for design on a system level:** This includes system-level requirements elicitation and analysis, functional analysis and allocation, and traceability down to discipline level.
- **They ensure proper communication and overview of the technical solution and trade-offs:** Through proper documentation and diagrams the systems engineer creates overview and ensures the team is on the same page.
- **They monitor risks and changes and prevent unforeseen errors:** Utilizing their vast experience in complex engineering projects, through proper risk management, reviews, and change control, the systems engineer exerts control over engineering efforts thereby reducing errors and defects.
- **They ensure proper testing and verification of the solution:** The systems engineer shall drive system-level acceptance testing and support testing activities at lower levels. They are also responsible for ensuring standards and regulations are adhered to.

The overall goal of the systems engineer is to strive for completeness, coherence, and integration of the complete technical solution. The systems engineer supports this through system thinking and technical leadership and they enhance the managerial project leadership from the PM, without replacing it.

To do so, the SE process is defined as a framework that is controlled and continuously improved. However, its application is flexible enough to be adapted to all different types of projects and technologies. Finally the process facilitates continuous improvement through various reviews and opportunities for retrospectives.

6.2. Defining Systems Engineering program tasks

The overall objectives of the SE function are to ensure that the system requirements are well defined, appropriate design choices and considerations are made, and the system is verified in terms of its initial requirements (either via testing or review). In tasks and activities that have a significant impact on implementation, the systems engineer has a visionary and guiding role.

It is possible to define the SE role's responsibilities by looking at the tasks that are performed over the course of the life cycle of a development project and determining which are 'system' tasks. Using the definitions in Chapter 3 (especially Section 3.4), the takeaways from the interviews, and the newly defined vision on SE it was possible to define the following holistic SE program:

1. **Perform solution synthesis:** Exploring the problem domain and coming up with a solution.
2. **Perform requirements and TPM analysis:** Determining and ranking critical requirements and TPMs.
3. **Perform functional analysis and allocation:** Creating a Functional and Allocated Baseline of the System.
4. **Create a Concept Design Description:** Creating a system level design description document that summarizes Concept Development.
5. **Create and maintain a System Architecture Overview:** Establishing and maintaining a single source-of-truth for use throughout the project.
6. **Support Design Verification and Validation Plan creation:** Creating a plan for design testing, verification, and validation.
7. **Support Synthesis, Analysis, Evaluation:** Supporting and monitoring of day-to-day design and synthesis, analysis, and evaluation activities over the system life cycle.
8. **Support Risk Management activities:** Coordinate and support System-level Risk planning, identification, analysis, handling, and monitoring over the system life cycle.
9. **Plan, coordinate, and conduct formal Design Review meetings:** Conduct design review meetings to evaluate results and identify problems and necessary actions.
10. **Lead integration and monitor verification activities:** Ensure system constituents properly work together and monitor verification (reviews and testing).
11. **Ensure proper documentation:** Ensure that Trade-Offs and engineering decisions are documented clearly and with rationale.

6.3. Updated Benchmark life cycle model

As discussed in Chapter 4 Existing internal processes, Benchmark employs its own Design Control Methodology Process [41]. As part of this thesis, the 7-step life cycle model was updated to include the above tasks, but it cannot be shown due to confidentiality reasons.

6.4. Defining implementation level-of-detail per project

At the start of the project, the PM shall indicate in the DDP to what level SE shall be performed, by listing what deliverables shall be generated during the project. Their decision shall be based on their own experience, but also on precedent projects and on initial risk, technical, and scope analyses.

6.5. Organisational structuring within Benchmark

It is proposed to establish a new competence group for SE, next to the existing groups for EE/ME/SW/Test. This allows for flexibility of project assignment of the systems engineers, as their involvement in a project should change as it progresses. A competence leader should be appointed to ensure continuity in the department and should become process owner of the SE process.

6.6. Conclusion

In conclusion, as per the vision defined in this chapter the SE role should provide top-down transdisciplinary technical leadership and serve to ensure proper balance between specification and execution, as well as the completeness, coherence, and integration of the technical solution. The SE program tasks were defined as a framework that can be controlled and continuously improved, while remaining flexible enough to be adapted to different types of projects and technologies.

In terms of SE program tasks, the main challenge lied in identifying the tasks that relate to the system as a whole. The systems engineer plays a crucial role in solution synthesis and takes on a leadership role as the project progresses. The overall objectives of the SE function are to ensure that the system requirements are well defined, appropriate design choices and considerations are made, and the system is validated in terms of its initial requirements.

This chapter also updated Benchmark's Design Control Methodology to include the SE program tasks, through which the role and responsibilities of the systems engineer in a Benchmark project becomes clearer. However, it remains challenging to formulate a single universally suitable level of detail for the implementation of these activities. The new SE process shall facilitate the PM in deciding to what level SE shall be performed, based on their own experience, precedent projects, and initial analyses.

7. BENCHMARK'S NEW SYSTEMS ENGINEERING PROCESS

This chapter introduces the new design of the SE procedure for Benchmark. This will replace the deprecated draft SE Process from 2014. It takes important input from Section 4.2 as this process will tie together the existing EE, ME, SW Processes. The following sections describe the newly designed process diagram, all encompassed deliverables, and briefly touch on some important tools for the SE.

7.1. Process diagram

As discussed in Section 4.2, next to an overarching design and development process, the engineering disciplines within Benchmark also each follow their own process. To tie these processes together, the specific SE activities and outputs over the project (design) life cycle are collected and grouped in a new process diagram that is similar to the existing EE and ME process diagrams. The diagram and detailed information about the project deliverables cannot be shown in this public version of the thesis due to confidentiality agreements. A shortened version is shown below:

Activities:

Kick-Off
Solution Synthesis
Requirements and Functional Analysis
Concept Development
Requirements Specification
Detailed Design
Design Review
Design Verification and Validation
TPD check
Design Release and start of Life Cycle Management

Deliverables:

1. Systems Engineering Plan (new)
2. Proposed Solution Description (new)
3. Product Requirements Specification
4. System Interface Specification (new)
5. Product Concept Description (new)
6. Risk Analysis Documentation
7. System Architecture Overview (new)
8. Product Design Description
9. Product Test Plan
10. Requirements Traceability Documentation
11. Product Configuration Plan (new)
12. Change Control Board Records
13. Systems Engineering Acceptance Report (new)
14. Product Test Report
15. Waivers
16. Product Certifications
17. Integration, Verification, Validation and Qualification Plan (new)
18. Future Improvements List (new)

7.2. Conclusion

This chapter presented the new Benchmark SE Process, connecting the EE, ME, SW Processes and the Engineering Design Control Methodology. The process consists of ten activities, namely *Kick-Off*, *Solution Synthesis*, *Requirements and Functional Analysis*, *Concept Development*, *Requirements Specification*, D*etailed Design*, *Design Review*, *Design Verification and Validation*, *TPD Check*, *Design Release and start of Life Cycle Management*. The activities are presented in the same way as for the discipline processes (Section 4.2), while also adhering to the SE Program tasks as defined in Section 6.2.

Next to the activities, this chapter presented all deliverables that fall under the responsibility of the SE function. These are all deliverables created by the systems engineer during the design process required for successful design, verification and validation, and acceptance by the customer. To create these deliverables and perform the activities, the systems engineer can use a variety of tools. For this, this chapter introduced briefly some tools that can be considered part of the SE Toolbox.

The new process concretizes the SE role, and sets clear responsibilities. The new SEP, Proposed Solution Document, PCD, and AO deliverables improve the themes of communication and collaboration in the integrated product team. The theme of traceability and documentation is improved by introducing specialized requirements engineering tooling, and by shifting responsibility for the PRS and traceability documentation to the systems engineer.

Next, more involvement of the SE function in Benchmark's technology development phase, via the concretized Solution Synthesis activity improves the themes of system architecting and trade-off studies. Finally, the introduction of the SIS/ISR, SEAR, and reinstated responsibility of the systems engineer improve the themes of system integration and verification and validation.

8. IMPLEMENTATION APPROACH AND NEXT STEPS

The analysis and synthesis presented by this thesis form the groundwork for improving Benchmarks SE capabilities. Part II of this thesis identified improvement areas and Chapters 6 and 7 presented a draft program. However, for a successful and lasting implementation of the program, it must be tested and likely updated. In essence, the 'design' of the SE program is now prototypical, but it itself must now enter the Verification and Validation phase.

Implementing any new way-of-working or methodology is a complex and challenging task for an organization. However, by following a structured approach and considering key factors such as change management and company culture, it is possible to successfully introduce the new SE program and see significant benefits for the organization.

Change management literature suggests the following steps for implementing any new process, methodology, or way-of-working [42] [43]:

1. Develop a clear vision and design the new SE program by performing an analysis of the current situation and identifying key areas for improvement (this thesis).
2. Conduct a change management assessment to understand the potential impact of the new program on the organization and identify potential resistance to change.
3. Create and execute a pilot project to test the new way-of-working. Establish clear success criteria for the pilot. Evaluate the pilot project and collect feedback to adjust and refine the program.
4. Develop an implementation plan, including training and support for staff, and establish a governance structure to ensure the new program is effectively implemented and sustained.
5. Continuously monitor, evaluate, and improve the program to ensure it remains effective and aligned with Benchmark's goals and objectives.

The next sections detail all of these points, with the goal of forming a stepping stone for Benchmark to go forward with.

8.1. Change Management Assessment

Conducting a Change Management Assessment involves identifying and understanding the potential impact of the change on the organization as a whole, and identifying in advance any resistance to the new program so it can be addressed. In essence, it shares its spirit with an engineering change assessment in that the goal is to envision how the change may bring unwanted consequences or resistance.

8.1.1. Stakeholders and potential obstacles

It is important to acknowledge the stakeholders that will be affected through implementation of the vision. Understanding who the stakeholders are and how they will be affected by the change enables us to anticipate potential resistance so it can be addressed proactively.

The change must be supported and understanding of its rationale must propagate top-down from Competence Leadership down to the PMs and engineers. As the change progresses through the organization, groups of change ambassadors will be formed at lower levels. These ambassadors will help drive action and, together, form a guiding coalition.

8.1.2. Change management communication

Communication should happen often, through multiple channels, and through the right (set of) people. Benchmark should introduce the change early, by announcing the pilot and the intentions behind the new SE way-of-working. The engineering department should be updated on the progress often.

Project teams can be informed through in-person presentations which allow for questions and discussion on how the change may affect the team, but there must be care not to make these meetings too long and tiresome (which may increase resistance). Broader updates of implementation can be sent through emails and in the newsletter, although these usually fail to reach full potential as they are often only skimmed through.

To fully propagate the rationale behind the new way-of-working a more thorough campaign is necessary, for instance through a new in-house course in the LMS. Lastly, future SEs should receive targeted training in their responsibilities and what tools and techniques they should apply.

8.2. Pilot

The new program can be tested effectively by applying it in a pilot project. Benchmark should establish clear success criteria for the pilot and evaluate its effectiveness and pitfalls before the SE program is rolled out further.

The original plan for this thesis included conducting a pilot project. The aim of the pilot was to test the feasibility and effectiveness of the new process before presenting the design to the organisation. However, as the thesis project matured, it was decided to exclude the detailed design and execution of the pilot from its scope due to the extensive amount of time required to conduct a representative project. The focus of this thesis shifted towards outlining the new process and the necessary steps for its successful implementation, rather than conducting a pilot project.

Nevertheless, here I present some pointers for Benchmark to take into account during the process of selecting and designing a pilot project:

- The project should be representative of the type of work the organization typically undertakes and should be of sufficient complexity to test the new SE program. It should be a design project of a new product with undefined system architecture, and it should encompass all three engineering disciplines (EE, ME, SW).
- The project should have a clear scope, objectives, and deliverables, and should be well-defined in terms of timelines and resources. Unexpected delays are important indicators for potential improvements. A meandering or otherwise unsure customer can be detrimental to pilot success and should be avoided. An already existing customer that is confident in Benchmarks abilities is best.
- The project should be appropriately staffed, with a mix of experienced and less experienced colleagues, so that the new program can be tested across different levels of the organization. The project should be of reasonable size and scope warranting a multidisciplinary team of around eight engineers.
- The SE in the project should be the best available and should wholeheartedly support the new SE program. In executing the pilot it is important that the team comes to understand the need for a SE in a complex project. For this it is paramount that the SE is highly capable and with thorough understanding of their responsibilities via the new program.
- While all project team members should accept and understand the principles of the methods to be introduced, especially the PM must fully support the new way-of-working and understand the rationale behind it. After all it is the PMs responsibility to conduct the pilot and to make sure it will be successful. Therefore the project shall be led by a PM that has sufficient experience.
- The scope and complexity of the changes to the status-quo should be adjusted carefully to match the ability of the pilot project to facilitate these changes. In other words, is not necessary to improve everything at once: the highest priority is to separate the SE role, responsibilities, and budget. Later the organization may focus on concretizing the SE toolbox. After all, until now, Benchmark survived even without the change.

8.3. Implementation, training, and support

To ensure buy-in from leadership and engineers, several trainings should be implemented. Future systems engineers should follow a SE Masterclass to familiarize themselves with their new role and explicit responsibilities. A SE Knowledgebase should be created to serve as a common reference point for systems engineers in how to apply the various tools and techniques. Lastly, for all internal stakeholders (project engineers, PMs, etc.) there should be a LMS course to introduce the changes in responsibilities and program tasks. Using these channels enables the organisation to propagate the motivation for change to all levels. Figure 9 presents an overview.

SE masterclass	LMS trainings	Wiki-like knowledgebase
Who? All systems engineers and system architects.	**Who?** All engineers from the Design Engineering department.	**Who?** Anyone within Design Engineering.
What? In-depth knowlegde about SE processes, syetem thinking, toolbox.	**What?** Working knowledge of how the SE role has been redefined, including shifted responsibilities.	**What?** Overview of processes (incl. process owners), responsibilities, tools (incl. tool owners), deliverables, best practices, etc. for reference.

Figure 9. Types of SE training

8.4. Monitoring and continuous improvement

Regular reviews should be conducted to assess the impact of the new program on the organization and its environment, including any unintended consequences. It is crucial to involve all stakeholders in the monitoring and evaluation process, so that their perspectives and feedback are taken into account.

8.5. Conclusion

The goal of this chapter was to introduce the change steps necessary for implementing the new SE process and way-of-working. While the full implementation of the change is beyond the scope of this thesis, this chapter gave pointers on what steps to be taken next and some important factors.

This chapter discussed the need for a detailed Change Management Assessment, to identify stakeholders and potential obstacles, and to define a communication approach for the change process. It was identified as most important that the rationale behind the change must propagate top-down from Competence Leadership down to the PMs and engineers. A thorough communication campaign consisting of LMS trainings, an SE master class, and in-person presentations should support this.

The need for a pilot project was underlined next. This is crucial in refining the SE process and ensuring its effectiveness. It will not only allow for substantiated updates to the SE process, but an additional benefit is that the pilot team will become a strong support group for change. It is important to plan for success in the pilot project and to carefully monitor and evaluate its effectiveness.

This chapter also discussed some pointers for implementation of the new program, including training and support, and steps for continuous improvement. Prospective systems engineers should be trained via a SE Masterclass, and best-practices should be recorded in an internal SE Knowledgebase. Other project engineers should receive training via the already established LMS. Finally, the implementation and effectiveness of the SE program should be monitored continuously so that it can be updated and improved where necessary.

9. CONCLUSION AND RECOMMENDATIONS

9.1. Conclusion

The goal of this thesis was to define a new SE process for Benchmark. To do so in a structured manner, a theoretical framework was established and existing Benchmark processes and SE capabilities were analysed. The learnings of these analyses formed a good foundation for the synthesis of the end results: a newly designed vision on SE, a list of SE program tasks that define the SE responsibilities, and a formal SE process that integrates with Benchmark's way-of-working.

First, it was first necessary to familiarize ourselves with the general product life cycle and the academic state-of-the-art of SE (Chapters 2 and 3). With this state-of-the-art it was possible to start a discussion with key stakeholders in the organisation. Meetings once every two months with a focus group consisting of system architects, competence leadership, and lead engineers, led to a new vision on SE (Section 6.1). The essence of the vision is the following:

"Systems engineers should be responsible for a design on system level, ensure proper communication and overview of the technical solution and trade-offs, monitor risks and changes to prevent unforeseen errors, and ensure proper testing and verification of the solution."

Parallel to the creation of this vision, a detailed study was performed into Benchmark's existing processes (Chapter 4). Analysis of the formalized engineering processes and QMS procedures showed that Benchmark employs a structured approach to product design and that some SE principles are already applied, although not explicitly.

This study was enhanced with a SE capability assessment of Benchmark to identify areas for improvement (Chapter 5). As part of this, six PMs were interviewed in an effort to elucidate to what extent the formal processes are followed and what areas of technical management could be improved. This, supplemented with personal experience in working in project teams, led to important input for the synthesis of the new SE program.

Identified as most important were the need for proper communication and collaboration, a concrete SE role in a project with responsibility for requirements and design, traceability and thorough documentation, tooling for tracking of requirements, the value of collaboration and top-down management by the SE, and that the systems engineer should have a role in technical management. The main process areas that were identified for improvement were those of Integrated Product Teams, Risk Management, Trade-Off Studies, Requirements Management, and System Architecting.

Combining these findings, Chapters 6 and 7 present the results: a holistic SE framework that redefines project responsibilities, and a formalized SE process with well-defined deliverables, respectively.

The defined vision and identified SE program tasks were discussed extensively in the bimonthly focus group discussions and the formalized SE process underwent a call-for-review according to Benchmark's review procedure. These measures ensure that the new process has inherent support from the most important stakeholders; an important step in the change management that is to come.

Finally, while a pilot and further change management was in the end deemed out of scope for this thesis, Chapter 8 presents the four steps for successful implementation: stakeholder and change management, testing and iteration via a pilot, key trainings for implementation, and the necessity for monitoring and continuous improvement.

In conclusion, this thesis presents a well-defined vision for SE within Benchmark, which is supported by a focus group of select internal stakeholders. Furthermore, the identified SE program tasks and formally defined SE process form the basis of the envisioned implementation, based on the academic state-of-the-art all the while taking advantage of the SE that already exists. These, together with an outline of the to-be-taken next steps, allow Benchmark to continue in the future with the implementation of SE as described in this thesis.

9.2. Recommendations

This thesis forms the groundwork for Benchmark's new SE implementation. It has formed a vision, redefined program tasks and responsibilities, and presents a reviewed SE program. However, the new SE program has not yet been put in practice, nor has it been presented outside of the focus group that helped shape the vision.

Future research should be performed to design and execute a pilot program to test the new SE program and how the shift in responsibilities impacts the design process. While it is difficult to quantitatively measure the success of the new program, it is important to execute a pilot to see whether the SE program is successful in reducing unforeseen errors and cost. Some pointers for the design and execution of this pilot are presented in Chapter 8.

Another important part of implementing the new SE program comes down to the communication and implementation strategy. While the thesis presents a preliminary change assessment in Chapter 8, further research should be performed on how to effectively educate engineers and other stakeholders on the new workflow. Especially existing systems engineers and architects will see a welcome concretization of their responsibilities, but it is paramount this happens in a structured and unconvoluted way.

Additional further work is also necessary to create templates for the systems engineers to follow. As stated in Section 4.2, the existing discipline processes refer to templates to define how the activities should be performed. While the new SE process includes activities and deliverables, similar to the existing discipline processes, supportive templates should be created. This thesis concretized the *what*, *why*, and *when*, but not the *how*. Further research should be done to create templates to make sure the outputs of the activities are controlled.

Next, it is interesting to investigate the business case of SE; something this thesis has not touched upon. While research and industry trends suggest a necessity of and shift towards SE within engineering organisations, it is unclear to what extent the new SE program presented in this thesis will increase monetary profits for Benchmark. While successful implementation will likely reduce errors and benefit the quality of the designed products, it could be interesting to calculate the precise return-on-investment for Benchmark.

Lastly, an eventual future direction for Benchmark may be to investigate MBSE. As stated in Section 3.5, MBSE presents a new approach to SE, possibly improving requirements completeness, consistency, and communication. However its implementation is challenging and requires more experience with SE, for which this thesis forms the groundwork.

REFERENCES

[1] Benchmark, “System Engineering Process, Rev. 0.4,” Benchmark, Almelo, 2014.

[2] S. Jackson, “Introducing Systems Engineering into a Traditional Commercial Organisation,” in *INCOSE International Symposium 6(1)*, 1996.

[3] B. de Landtsheer, L. Aartman, J. Jamar, J. de Liefde, N. Malotaux, H. Reinhoudt and P. Schreinemakers, “Implementing Systems Engineering: A Step-By-Step Guide,” in *Fifth European Systems Engineering Conference*, Edinburgh, UK, 2006.

[4] B. S. Blanchard and W. J. Fabrycky, Systems Engineering and Analysis: Pearson New International Edition, 5th ed., London, United Kingdom: Pearson Education Limited, 2013.

[5] M. G. Bonnema, K. T. Veenvliet and J. F. Broenink, Systems Design and Engineering: Facilitating Multidisciplinary Development Projects, Boca Raton, FL: Taylor & Francis Group, 2016.

[6] ISE/IEC/IEEE, “ISO/IEC/IEEE 15288:2015 Systems and software engineering — System life cycle processes,” International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), Geneva, Switzerland, 2015.

[7] B. Dunbar, “NASA Program/Project Life Cycle,” National Aeronautics and Space Administration (NASA), 12 December 2019. [Online]. Available: https://www.nasa.gov/seh/3-project-life-cycle. [Accessed 1 June 2022].

[8] United States of America Department of Defence, “DoD Instruction 5000.85 Major Capability Acquisition,” Office of the Under Secretary of Defense for Acquisition and Sustainment, Washington, DC, 2020.

[9] INCOSE, INCOSE Systems Engineering Handbook, 4th ed., San Diego, CA, USA: International Council on Systems Engineering (INCOSE), 2015.

[10] D. Fairley, K. Forsberg, R. Madachy and P. Marbach, “System Lifecycle Process Models: Vee,” Systems Engineering Book of Knowledge (SEBoK), 22 May 2022. [Online]. Available: https://www.sebokwiki.org/wiki/System_Lifecycle_Process_Models:_Vee#Development_Stage. [Accessed 3 June 2022].

[11] K. Forsberg, R. Turner and R. Adcock, “Generic Life Cycle Model,” Systems Engineering Book of Knowledge, [Online]. Available: https://www.sebokwiki.org/wiki/Generic_Life_Cycle_Model. [Accessed 5 July 2022].

[12] G. Altshuller, 40 Principles: TRIZ Keys to Innovation, Technical Innovation Center, Inc., 2002, 2002.

[13] Systematic Inventive Thinking, “Systematic Inventive Thinking,” Systematic Inventive Thinking, [Online]. Available: https://www.sitsite.com/. [Accessed 2 June 2022].

[14] M. Sridharan, “Pugh Matrix,” Think Insights, [Online]. Available: https://thinkinsights.net/consulting/pugh-matrix/. [Accessed 2 June 2022].

[15] R. Vongvit, “The TRIZ Integrated PUGH Model Approach for Design Problem Solving,” in *5th International Conference on Information Management (ICIM)*, Cambridge, UK, 2019.

[16] K. Wurthmann, “Conducting Pugh Method-based Trade Studies during Product Development: The case of evaluating Turbofan versus Turboprop versus Piston Engine Alternatives for UAVs,” in *AIAA Scitech 2020 Forum*, Orlando, FL, 2020.

[17] G. M. Bonnema, “Function and budget based system architecting,” in *6th International Symposium on Tools and Methods of Competitive Engineering, TMCE 2006*, Ljubljana, Slovenia, 2006.

[18] G. M. Bonnema, “FunKey Architecting: an integrated approach to system architecting using functions, key drivers and system budgets,” University of Twente, Enschede, The Netherlands, 2008.

[19] QFD Institute, “Quality Function Deployment,” QFD Institute, [Online]. Available: http://qfdi.org/. [Accessed 02 June 2022].

[20] C. Govers, “What and how about quality function deployment (QFD),” *International Journal of Production Economics,* Vols. 46-47, pp. 575-585, 1996.

[21] Electronic Industries Alliance, “ANSI/EIA-632-1999 Processes for Engineering a System,” Electronic Industries Alliance, Arlington, VA, USA, 1999.

[22] INCOSE, “Systems Engineering,” International Council on Systems Engineering (INCOSE), [Online]. Available: https://www.incose.org/about-systems-engineering/system-and-se-definition/systems-engineering-definition. [Accessed 12 May 2022].

[23] H. W. Bode, “Report to the Committee on Science and Astronautics,” U.S. House of Representatives, Washington, 1967.

[24] National Aeronautics and Space Administration (NASA), “Fundamentals of Systems Engineering,” [Online]. Available: https://www.nasa.gov/seh/2-fundamentals. [Accessed 12 May 2022].

[25] ISO/IEC/IEEE, “Systems and Software Engineering - System and Software Engineering Vocabulary (SEVocab),” International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), Geneva, Switzerland, 2010.

[26] H. G. Sillitto, R. Griego, E. Arnold, D. Dori, J. Martin, P. Godfrey, D. Mckinney, D. Krob and S. Jackson, “A fresh look at Systems Engineering - what is it, how should it work?,” in *INCOSE International Symposium 28(1):955-970*, 2018.

[27] S. Jackson, B. Botha and A. Boglo, “Three Views of Systems Engineering,” *Project Performance International publication SyEN,* vol. 89, 22 May 2020.

[28] B. Boehm and J. W. Hansen, “Spiral Development: Experience, Principles, and Refinements Spiral Development Workshop February 9, 2000,” Software Engineering Institute, Los Angeles, 2000.

[29] L. Osborne, J. Brummond, R. Hart, M. Zarean and S. Conger, “Clarus Concept of Operations,” Federal Highway Administration, 2005.

[30] S. A. Sheard, “Three Types of Systems Engineering Implementation,” in *Proceedings of the Tenth International Symposium of the International Council on Systems Engineering*, Minneapolis, 2000.

[31] International Organisation for Standardisation (ISO), “Quality Management Systems - Requirements (ISO 9001-2015),” ISO, Geneva, 2015.

[32] R. J. Elm, R. D. Goldenson, E. K. Emam, N. DOnatelli and A. Neisa, “A Survey of Systems Engineering Effectiveness,” National Defense Industrial Association, Hanscom, MA, 2008.

[33] CMMI Product Team, “CMMI for Systems Engineering/Software Engineering,” Software Engineering Institute, Pittsburgh, PA, 2001.

[34] N. Shevchenko, “An Introduction to Model-Based Systems Engineering (MBSE),” Carnegie Mellon University: Software Engineering Institute, 21 December 2022. [Online]. Available: https://insights.sei.cmu.edu/blog/introduction-model-based-systems-engineering-mbse/. [Accessed 2022 May 20].

[35] A. M. Madni and M. Sievers, “Model-based systems engineering:Motivation, current status, and research opportunities,” *Systems Engineering,* vol. 21, pp. 172-190, 2018.

[36] Benchmark, “Electronics Engineering Process, Rev. 4.0,” Benchmark, Almelo, 2021.

[37] Benchmark, “Software Engineering Process, Rev. E,” Benchmark, Global, unknown.

[38] Benchmark, “Mechanical Engineering Process, Rev. 3.0,” Benchmark, Almelo, 2015.

[39] P. J. Elm and R. D. Goldenson, “The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey,” National Defense Industrial Association, Hanscom, MA, 2012.

[40] Software Engineering Institute (SEI), “Capability Maturity Model Integrated (CMMI) for Development,” Software Engineering Institute (SEI)/Carnegie Mellon University (CMU), Pittsburgh, PA, 2002.

[41] Benchmark, “Engineering Design Control Methodology, Rev. H,” Benchmark, Global, 2022.

[42] J. P. Kotter, Leading Change, Boston, MA, USA: Harvard Business Press, 2012.

[43] J. Kotter and D. Cohen, The Heart of Change: Real-Life Stories of How People Change Their Organisations, Cambridge, MA: Harvard Business School Press, 2002.

[44] M. Watson, B. Masmer, G. Roedler, D. Rousseau, J. Calvo-Amodio, C. Keating, W. D. Miller, S. Lucero, R. Gold, C. Jones, D. Long, R. W. Russel and A. Sedmak, Systems Engineering Principles, San Diego, CA: INCOSE, 2022.

[45] A. Viken and G. Muller, "Creating and Applying A3 Architecture Overviews: A Case Study in Software Development," *INCOSE International Symposium,* vol. 28, no. 1, pp. 378-391, 2018.

[46] W. Pesselse, T. Hofman, M. Simons and G. Muller, "Applying and analyzing A3 Architecture Overviews in a complex and dynamic engineering environment," *INCOSE International Symposium,* vol. 29, no. 1, pp. 952-970, 2019.

[47] J. P. Kotter and H. Rathgenber, Out Iceberg is Melting: Changing and Succeeding Under Any Conditions, Pan Macmillan UK, 2017.

[48] E. C. Honour, "Systems Engineering return on investment," Defense and Systems Institute, University of South Wales, 2013. [Online]. Available: https://www.hcode.com/seroi/documents/SE-ROI%20Thesis-distrib.pdf. [Accessed 29 November 2022].

[49] D. de Groot, B. Breazu, P. Verhoef and A. Zambon, "A3AO of The Self-Propelling Caravan: Allowing EVs to Effectively tow a Caravan for Holidays," University of Twente, Enschede, 2022.

[50] S. Burge, "The Systems Engineering Tool Box: Needs Means Analysis (NMA)," Burge Hughes Walsh, Warwickshire, UK, 2003.

[51] P. J. Borches, "A3AO Showcase," University of Twente, 2018. [Online]. Available: https://a3architecture-9022e.firebaseapp.com/showcase.html. [Accessed 2 January 2023].

[52] P. J. Borches, "A3AO Applications," University of Twente, 2018. [Online]. Available: https://a3ao.eu/. [Accessed 2 January 2023].

[53] P. J. Borches, "A3 Architecture overviews: A tool for effective communication in product evolution," University of Twente, Enschede, 2010.

[54] T. Boge and K. Falk, "A3 Architecture Views – A Project Management Tool?," *INCOSE International Symposium,* vol. 29, no. 1, pp. 971-987, 2019.

[55] Y. Akao, Quality Function Deployment, Cambride, MA: Productivity Press, 1990.

[56] J. Snoderly and A. Faisandier, "System Verification," Systems Engineering Body of Knowledge (SEBoK), 31 October 2022. [Online]. Available: https://www.sebokwiki.org/wiki/System_Verification#Verification_versus_Validation. [Accessed 25 November 2022].

[57] Department of Defense (DoD), "Risk, Issue, and Opportunity Management Guide for Defense Acquisition Programs," USA: Office of the Deputy Assistant Secretary of Defense for Systems Engineering/Department of Defense, Washington, DC, 2015.

[58] Project Management Institute (PMI), "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)," Project Management Institute (PMI), Newtown Square, PA, 2013.

[59] P. Malik, "The Kübler Ross Change Curve in the Workplace (2023)," Whatfix, 24 February 2022. [Online]. Available: https://whatfix.com/blog/kubler-ross-change-curve/#:~:text=The%20Change%20Curve%2C%20or%20Kübler,bargaining%2C%20depression%2C%20and%20acceptance.. [Accessed 30 January 2023].

[60] V. V. Souchkov, "TRIZ & Systematic Innovation: Advanced TRIZ and xTRIZ: Collection of Artilcles and References," ICG Training & Consulting, Enschede, The Netherlands, 2017.

[61] H. C. Theisens, Climbing the mountain: Lean Six Sigma Green Belt, Enschede, The Netherlands: Lean Six Sigma Academy, 2021.

[62] Benchmark, "Medical Risk Management Procedure, Rev. F," Benchmark, Global, 2020.

APPENDIX

A — EE Process (redacted)

B — ME Process (redacted)

C — SEC Survey (results redacted)

D — Interview Transcripts (redacted)

E — SE Process document (redacted)

APPENDIX C — SEC SURVEY

Survey questions and scoring procedure

The table below shows the survey questions including how they are linked to the SE process areas, the response type, and assessment of the response.

For each assessment criteria (SE process areas ***IPT***, ***PP***, ***PMC***, ***RSK***, ***REQ***, ***TRD***, ***ARC***, ***PI***, ***VER***, ***VAL***, ***CM***, and general ***PERF***, ***PC***), the assessed values are combined into a weighted summed index to create the assessment. The final scores are scaled from 1 (very low capability) to 4 (very high capability). General info like project market and development team size was also collected.

Id.	Question	Performance PERF	Integrated Product Teams IPT	Project Planning PP	Project Monitoring and Control PMC	Risk mgmt RSK	Requirements mgmt REQ	Trade-off Studies TRD	Architecting ARC	Integration PI	Verification VER	Validation VAL	Configuration mgmt CM	Project Challenge PC	Response type	assessment
A01	The project is challenging because there is no precedent for what is being done.													x	agree/disagree	1 = disagree, 4 = agree
A02	The project is challenging because significant constraints are placed on the quality attributes of the product.													x	agree/disagree	1 = disagree, 4 = agree
A03	The project is challenging because the size of the development effort is large.													x	agree/disagree	1 = disagree, 4 = agree
A04	The project is challenging because the technology needed for this project is not mature or otherwise poses a high risk.													x	agree/disagree	1 = disagree, 4 = agree
A05	The project is challenging because there are extensive needs for interoperability with other systems.													x	agree/disagree	1 = disagree, 4 = agree
A06	The project is challenging because there are insufficient resources available to support the project.													x	agree/disagree	1 = disagree, 4 = agree
A07	The project is challenging because there are insufficient skills and subject matter expertise available to support the project.													x	agree/disagree	1 = disagree, 4 = agree
A09	In the past, this project team has NOT successfully completed projects of similar scope.													x	agree/disagree	1 = disagree, 4 = agree
C01	In the past, Benchmark has NOT successfully completed projects of similar scope.													x	agree/disagree	1 = disagree, 4 = agree
A10	The requirements supplied by the customer for this project are NOT well-defined.													x	agree/disagree	1 = disagree, 4 = agree
A11	In executing the project, the requirements supplied by the customer for this project have NOT changed sufficiently to generate a significant impact on the project.													x	agree/disagree	1 = disagree, 4 = agree
A12	What percentage of the customer technical requirements were marked “TBD” or equivalent at time of contract award?													x	%	1 ⟵ 5% > (answer) 2 ⟵ 10% > (answer) ≥ 5% 3 ⟵ 20% > (answer) ≥ 10% 4 ⟵ (answer) ≥ 20%

Id.	Question	PERF	IPT	PP	PMC	RSK	REQ	TRD	ARC	PI	VER	VAL	CM	PC	Response type	Assessment
A13	What percentage of the customer's technical requirements are currently marked "TBD" or equivalent?													x	%	1 ⟵ 5% > (answer) 2 ⟵ 10% > (answer) ≥ 5% 3 ⟵ 20% > (answer) ≥ 10% 4 ⟵ (answer) ≥ 20%
A14	Do you separately budget and track SE activities?			x	x										y/n/?	4=Yes, 1=No, 1=?
B01	What is the current total contract value of this project?	x												x	€	determine based on received answers
B02	What was the initial contract value of this project?	x												x	€	determine based on received answers
B03	The change in contract value is primarily due to	x													dropdown	1⟵N/A, 2⟵change in tech/scope, 3⟵unplanned in-creases, 4⟵other
B07	What is the current total budget for this project?	x												x	€	determine based on received answers
B08	What was the initial total budget for this project?	x												x	€	determine based on received answers
B09	The change in budget is primarily due to	x													dropdown	1⟵N/A, 2⟵change in tech/scope, 3⟵unplanned in-creases, 4⟵customer driven, 5⟵other
O09	I believe that my customer is satisfied with this project wit respect to cost	x													agree/disagree	1 = disagree, 4 = agree
B04	What is the current total planned duration of this project or contract?	x												x	months	determine based on received answers
B05	What was the initial total planned duration of this project or contract?	x												x	months	determine based on received answers
B06	The change in schedule is primarily due to	x													dropdown	1⟵N/A, 2⟵change in tech/scope, 3⟵unplanned in-creases, 4⟵customer driven, 5⟵other
N08	What is the projected schedule variance at completion for the current contract baseline?	x		x											months	determine based on received answers
O08	I believe that my customer is satisfied with this project's performance with respect to the schedule.	x		x											agree/disagree	1 = disagree, 4 = agree
D01	This project utilizes/utilized a documented set of SE processes for the planning and execution of the project.			x											agree/disagree	1 = disagree, 4 = agree
D02	This project has/had an accurate and up-to-date WBS that included task descriptions and work package descrip-tions.			x											agree/disagree	1 = disagree, 4 = agree
D03	This project has/had an accurate and up-to-date WBS that was based on the product structure.			x											agree/disagree	1 = disagree, 4 = agree
D04	This project has/had an accurate and up-to-date WBS that was developed with the active participation of those who perform the systems engineering activities.			x											agree/disagree	1 = disagree, 4 = agree
D05	This project has/had an accurate and up-to-date WBS that was developed and maintained with the active participa-tion of all relevant stakeholders.			x											agree/disagree	1 = disagree, 4 = agree
D06	This project's Technical Approach (e.g. hardware, software, mechanical development plan) is tailored to the project.			x											agree/disagree	1 = disagree, 4 = agree

Id.	Question	PERF	IPT	PP	PMC	RSK	REQ	TRD	ARC	PI	VER	VAL	CM	PC	Response type	Assessment
D07	This project's Technical Approach (e.g. hardware, software, mechanical development plan) is complete, accurate and up-to-date.			x											agree/disagree	1 = disagree, 4 = agree
D08	This project's Technical Approach (e.g. hardware, software, mechanical development plan) is developed and maintained with the active participation of those who perform the Systems Engineering activities.			x											agree/disagree	1 = disagree, 4 = agree
D09	This project's Technical Approach (e.g. hardware, software, mechanical development plan) is developed and maintained with the active participation of all appropriate functional stakeholders.			x											agree/disagree	1 = disagree, 4 = agree
D10	This project has a top-level plan, such as an IMP, that is an event-driven plan (i.e., each accomplishment is tied to a key project event).			x											agree/disagree	1 = disagree, 4 = agree
D11	This project has a top-level plan, such as an IMP, that documents significant accomplishments with pass/fail accomplishment criteria for both business and technical elements of the project.			x											agree/disagree	1 = disagree, 4 = agree
D12	This project has a top-level plan, such as an IMP, that is consistent with the WBS.			x											agree/disagree	1 = disagree, 4 = agree
D13	This project has an integrated event-based schedule that is structured as a networked, multi-layered schedule of project tasks required to complete the work effort.			x											agree/disagree	1 = disagree, 4 = agree
D14	This project has an integrated event-based schedule that contains a compilation of key technical accomplishments (e.g., a SE Master Schedule).			x											agree/disagree	1 = disagree, 4 = agree
D15	This project has an integrated event-based schedule that references measurable criteria (usually contained in the IMP) required for successful completion of key technical accomplishments.			x											agree/disagree	1 = disagree, 4 = agree
D16	This project has an integrated event-based schedule that is consistent with the WBS.			x											agree/disagree	1 = disagree, 4 = agree
D17	This project has an integrated event-based schedule that identifies the critical path of the program schedule.			x											agree/disagree	1 = disagree, 4 = agree
D18	This project has a plan or plans for the performance of technical reviews with defined entry and exit criteria throughout the lifecycle of the project.			x											agree/disagree	1 = disagree, 4 = agree
D19	The SE function actively participates in the development and updates of the project planning.			x											agree/disagree	1 = disagree, 4 = agree
D20	Those who perform SE activities actively participate in tracking/reporting of task progress.			x	x										agree/disagree	1 = disagree, 4 = agree
D21	This project has a plan or plans that include details of the management of the integrated technical effort across the project (e.g., a SE Mgt. Plan or a SE Plan).			x											agree/disagree	1 = disagree, 4 = agree
E01	This project makes effective use of IPTs.		x												agree/disagree	1 = disagree, 4 = agree
E02	My client participates in my IPTs for this project.		x												agree/disagree	1 = disagree, 4 = agree
E03	My suppliers actively participate in my IPTs for this project.		x												agree/disagree	1 = disagree, 4 = agree
E04	This project has an IPT with assigned responsibility for SE.		x												agree/disagree	1 = disagree, 4 = agree
E05	This project has SE representation on each IPT.		x												agree/disagree	1 = disagree, 4 = agree
F01	This project has a Risk Management process that creates and maintains an accurate and up-to-date list of risks affecting the project.					x									agree/disagree	1 = disagree, 4 = agree
F02	This project has a Risk Management process that creates and maintains up-to-date documentation of risk mitigation plans and contingency plans for selected risks.					x									agree/disagree	1 = disagree, 4 = agree
F03	This project has a Risk Management process that monitors and reports the status of risk mitigation activities and resources.					x									agree/disagree	1 = disagree, 4 = agree
F04	This project has a Risk Management process that assesses risk against achievement of an event-based schedule.					x									agree/disagree	1 = disagree, 4 = agree
F05	This project's Risk Management process is integrated with project decision-making.					x									agree/disagree	1 = disagree, 4 = agree

Id.	Question	PERF	IPT	PP	PMC	RSK	REQ	TRD	ARC	PI	VER	VAL	CM	PC	Response type	Assessment
F06	This project's Risk Management process is integrated with program cost and/or earned value management.					x									agree/disagree	1 = disagree, 4 = agree
F07	This project's Risk Management process is integrated with program scheduling.					x									agree/disagree	1 = disagree, 4 = agree
F08	This project's Risk Management process integrates subcontract or supplier risk management processes.					x									agree/disagree	1 = disagree, 4 = agree
	This project has high development/technological-readiness risk.					x								x		
	This medical project has high patient risk. [Leave blank if not medical project.]					x								x		
G01	This project maintains an up-to-date and accurate listing of all requirements specified by the customer, to include regulatory, statutory, and certification requirements.						x								agree/disagree	1 = disagree, 4 = agree
G02	This project maintains an up-to-date and accurate listing of all requirements derived from those specified by the customer.						x								agree/disagree	1 = disagree, 4 = agree
G03	This project maintains up-to-date and accurate documentation clearly reflecting the hierarchical allocation of both customer and derived requirements to each element (subsystem, component, etc.) of the system in the configuration baselines.						x								agree/disagree	1 = disagree, 4 = agree
G04	This project documents and maintains accurate and up-to-date descriptions of operational concepts and their associated scenarios.						x								agree/disagree	1 = disagree, 4 = agree
G05	This project documents and maintains accurate and up-to-date descriptions of use cases (or their equivalent).						x								agree/disagree	1 = disagree, 4 = agree
G06	This project documents and maintains accurate and up-to-date descriptions of product installation, maintenance and support concepts.						x								agree/disagree	1 = disagree, 4 = agree
G08	This project has documented criteria (e.g., cost impact, schedule impact, authorization of source, contract scope, requirement quality) for evaluation and acceptance of requirements.						x								agree/disagree	1 = disagree, 4 = agree
G09	The requirements for this project are approved in a formal and documented manner by relevant stakeholders.						x								agree/disagree	1 = disagree, 4 = agree
G10	This project performs and documents requirements impact assessments for proposed requirements changes.						x								agree/disagree	1 = disagree, 4 = agree
G11	This project develops and documents project requirements based on stakeholder needs, expectations, and constraints.						x								agree/disagree	1 = disagree, 4 = agree
G12	This project has an accurate and up-to-date requirements management system.						x								agree/disagree	1 = disagree, 4 = agree
G13	For this project, the requirements documents are managed under a configuration control process.						x						x		agree/disagree	1 = disagree, 4 = agree
G14	For this project, the requirements documents are accessible to all relevant project staff.						x								agree/disagree	1 = disagree, 4 = agree
O03	Requirements are being satisfied and remain on track to be satisfied in the product releases as originally planned. They are not being deleted or deferred to later releases.	x					x								agree/disagree	1 = disagree, 4 = agree
O10	I believe that my customer is satisfied with this project's performance with respect to satisfaction of requirements	x					x								agree/disagree	1 = disagree, 4 = agree
H01	Stakeholders impacted by trade-off studies are involved in the development and performance of those trade-off studies.							x							agree/disagree	1 = disagree, 4 = agree
H02	This project performs and documents trade-off studies between alternate solutions in a timely manner, and based on definitive and documented selection criteria.							x							agree/disagree	1 = disagree, 4 = agree
H03	Documentation of trade-off studies is maintained in a defined repository and is accessible to all relevant project staff.							x							agree/disagree	1 = disagree, 4 = agree
I01	This project maintains accurate and up-to-date descriptions (e.g. interface control documents, models, etc.) defining interfaces in detail.								x						agree/disagree	1 = disagree, 4 = agree
I02	Interface definition descriptions are maintained in a designated location, under configuration management, and accessible to all who need them.								x						agree/disagree	1 = disagree, 4 = agree

Id.	Question	PERF	IPT	PP	PMC	RSK	REQ	TRD	ARC	PI	VER	VAL	CM	PC	Response type	Assessment
I03	For this project, the product high-level structure is documented, kept up to date, and managed under configuration control.								x						agree/disagree	1 = disagree, 4 = agree
I04	For this project, the product high-level structure is documented using multiple views (e.g. functional views, module views, etc.).								x						agree/disagree	1 = disagree, 4 = agree
I05	For this project, the product high-level structure is accessible to all relevant project personnel.								x						agree/disagree	1 = disagree, 4 = agree
J01	This project has accurate and up-to-date documents defining its product integration process, plans, criteria, etc. throughout the lifecycle.									x					agree/disagree	1 = disagree, 4 = agree
K01	This project has accurate and up-to-date documents defining the procedures used for the test and verification of systems and system elements.										x				agree/disagree	1 = disagree, 4 = agree
K02	This project has accurate and up-to-date documents defining acceptance criteria used for the verification of systems and system elements.										x				agree/disagree	1 = disagree, 4 = agree
K03	This project has a documented and practiced review process for work packages that defines entry and exit criteria.										x				agree/disagree	1 = disagree, 4 = agree
K04	This project has a documented and practiced review process that includes training the reviewers to conduct reviews.										x				agree/disagree	1 = disagree, 4 = agree
K05	This project has a documented and practiced review process that defines criteria for the selection of work packages for review.										x				agree/disagree	1 = disagree, 4 = agree
K06	This project has a documented and practiced review process that tracks action items to closure.				x						x				agree/disagree	1 = disagree, 4 = agree
K07	This project has a documented and practiced review process that addresses identified risks and risk mitigation activ-ities during reviews.										x				agree/disagree	1 = disagree, 4 = agree
K08	This project has a documented and practiced review process that examines completeness of configuration base-lines.										x				agree/disagree	1 = disagree, 4 = agree
K09	This project conducts reviews and documents results, issues, action items, risks, and risk mitigations.										x				agree/disagree	1 = disagree, 4 = agree
L01	This project has accurate and up-to-date documents defining the procedures used for the validation of systems and system elements.											x			agree/disagree	1 = disagree, 4 = agree
L02	This project has accurate and up-to-date documents defining acceptance criteria used for the validation of systems and system elements.											x			agree/disagree	1 = disagree, 4 = agree
M01	This project maintains a listing of items managed under configuration control.												x		agree/disagree	1 = disagree, 4 = agree
M02	This project has a configuration management system that charters a CCB to disposition change requests.												x		agree/disagree	1 = disagree, 4 = agree
M03	This project maintains records of requested and implemented changes to configuration-managed items.												x		agree/disagree	1 = disagree, 4 = agree
M04	This project creates and manages configuration baselines.												x		agree/disagree	1 = disagree, 4 = agree
N01	This project creates and manages cost and schedule baselines.				x										agree/disagree	1 = disagree, 4 = agree
N02	EVA data are available to decision makers in a timely manner.				x										agree/disagree	1 = disagree, 4 = agree
N04	Variance thresholds for the CPI and SPI are defined, documented, and used to determine when corrective action is needed.				x										agree/disagree	1 = disagree, 4 = agree
N05	The EVA data are linked to the technical effort through the WBS, and the IMS.				x										agree/disagree	1 = disagree, 4 = agree
O06	This project collects and tracks (or will collect and track) customer PRs.				x										agree/disagree	1 = disagree, 4 = agree
O07	This project conducts (or will conduct) engineering assessments of all customer PRs.				x										agree/disagree	1 = disagree, 4 = agree