Implementing Systems Engineering in New Product Development at Benchmark Electronics

David de Groot

MSc Thesis 16-Mar-23

University of Twente

Benchmark Electronics

Confidential until 28-Mar-2024

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)

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. SE is vital for creating high-quality products that meet the needs of the customer 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 develop have the potential to save lives, improve quality of life, or enhance productivity. It feels honourable to contribute to society with these products, 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. Next, this project would not have been possible without the support and collaboration of the entire Benchmark team. Therefore, I would 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 SE 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 for companies like Benchmark 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 adapted to its project life cycle model and compatible with the company's existing engineering processes.

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 program 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 Electrical Engineering, Mechanical Engineering, and Software 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
DDP	Design and Development Plan
DFM	Design for Manufacturing
DFMEA	Design Failure Modes and Effects Analysis
DFT	Design for Testing
DHF	Design History File
DMS	Document Management System
DVP	Design Verification Plan
DVT	Design Verification Testing
ECD	Electronics Concept Description
EE	Electrical Engineering
EMC	Electromagnetic Compatibility
ERS	Electronics Requirements Specification
ESD	Electrostatic Discharge
EVA	Earned Value Analysis
FDA	Food and Drug Association
FMEA	Failure Modes and Effects Analysis
FTE	Full-Time Equivalent
IPO	Input-Process-Output
IPT	Integrated Product Team
IRS	Interface Requirements Specification
IVVQ	Integration, Verification, Validation, and Qualification Plan
LMS	Learning Management System
MBSE	Model-Based Systems Engineering
ME	Mechanical Engineering
MRS	Mechanical Requirements Specification
MTP	Mechanical Test Plan
NPI	New Product Introduction
PCB	Printed Circuit Board
PCD	Product Concept Description
PDD	Product Design Description
PM	Program Manager
PRS	Product Requirements Specification
PTP	Product Test Plan
QFD	Quality Function Deployment
QMS	Quality Management System
R&D	Research and development
SE	Systems Engineering
SEAR	Systems Engineering Acceptance Report
SEP	Systems Engineering Plan
SIS	System Interface Specification
SRS	Software Requirements Specification
SRS	Software Requirements Specification
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
WBS	Work Breakdown Structure

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 a 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 to perform well, 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 diagram 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."

Then, to supplement Benchmark's life cycle model, the SE program tasks within the system life cycle were identified. These form the responsibilities of the SE function within a project. The following tasks were defined:

1. *Perform Solution Synthesis*
2. *Perform Requirements and TPM 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 new 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 key internal stakeholders. The identified SE program tasks and formally defined SE process form the foundation of the envisioned implementation, which is based on academic state-of-the-art while also leveraging the SE that already exists. These, along with an outline of the next steps to be taken, 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 OEMs 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, serves Benchmark's goal 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 Research and Development (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 really 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 becoming increasingly 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 (partly) by 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 commissioned by the NHS. Tests up to 4 people, on-the-spot, at the same time.	*Combines a wearable device for new-borns to monitor neonatal vital signs with IOT and a wireless monitor.*	*Point-of-care microfluidic analyser for a variety of tests for infectious and cardiovascular disease, immunology, drug monitoring, etc.*
		
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, multi-load monitoring, data logging, web interface, etc. Enabling safe and accurate power measurement.*

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.

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, a formalized process is currently not in place. Because of this, it is often unclear where the responsibilities of the Systems Engineer begin and end. This 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 primarily 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 analyse the current situation, create a vision, and to design a strategy for implementation.

Learnings discussed by S. Jackson [2] and De Landtsheer et al. [3] have informed the approach used in this thesis 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 process that is presented in Chapter 7. Finally, although the pilot testing of this process 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 SE and lays out the various academic and industry definitions of SE. Because there are many available, common threads are identified in order to establish a single definition for the remainder of the thesis. 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.

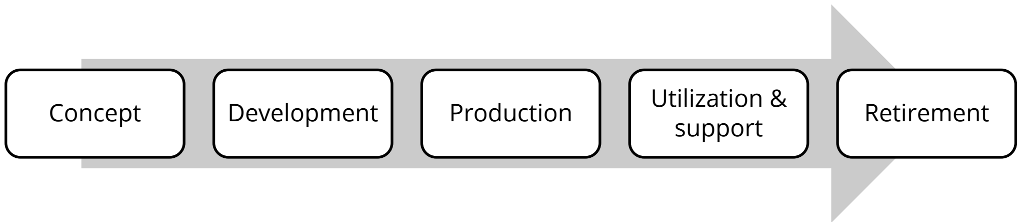


Figure 1. Generic system life cycle

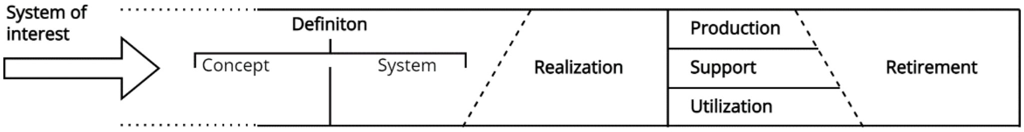
While the life cycles stages themselves can be defined differently per organisation, the overarching idea is the same: as stated excellently in the INCOSE Systems Engineering Handbook, "the needs of each subsequent life cycle stage must be considered during the earlier stages … in order to make the appropriate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

Figure 2 on the next page 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 come from recognized authorities in the field of SE who 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.

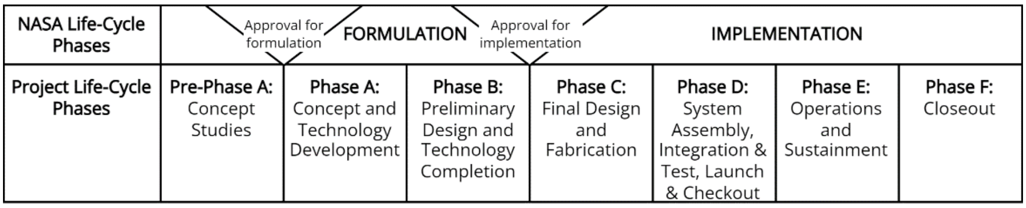
The next sections then give an overview of the generic life cycle model phases distilled from these sources and the activities therein. They combine the learnings from the above sources into one a generic life cycle model description over Sections 2.1-2.6, to serve as the basis for understanding the system life cycle.

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

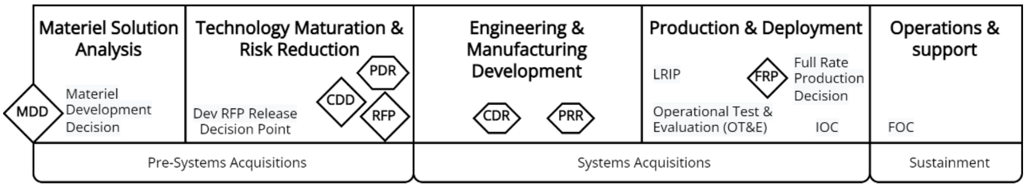
A. Generic life cycle model published in ISO 15288:2015 (figure reproduced from [6])



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

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 Technology Readiness Level (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 Theory of Inventive Problem Solving (TRIZ) [12], Systematic Inventive Thinking [13], morphological charting, financial forecasting, Pugh comparison [14] [15] [16], FunKey Architecting [17] [18], Quality Function Deployment (QFD) [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 Failure Modes and Effects Analysis (FMEA) 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, computer-aided-design (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, Model-Based SE (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).

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. Definitions from INCOSE, NASA, and ISO/IEC/IEEE are listed on the next page.

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].

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 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

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

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 0).

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.

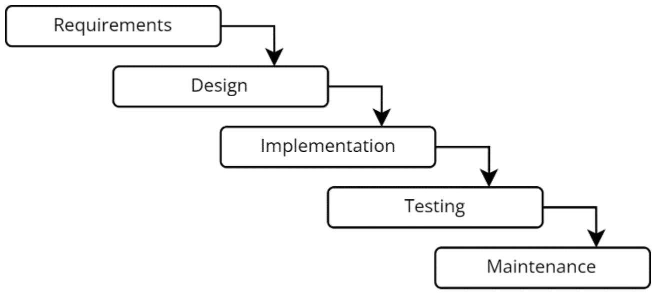


Figure 3. Waterfall model

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.

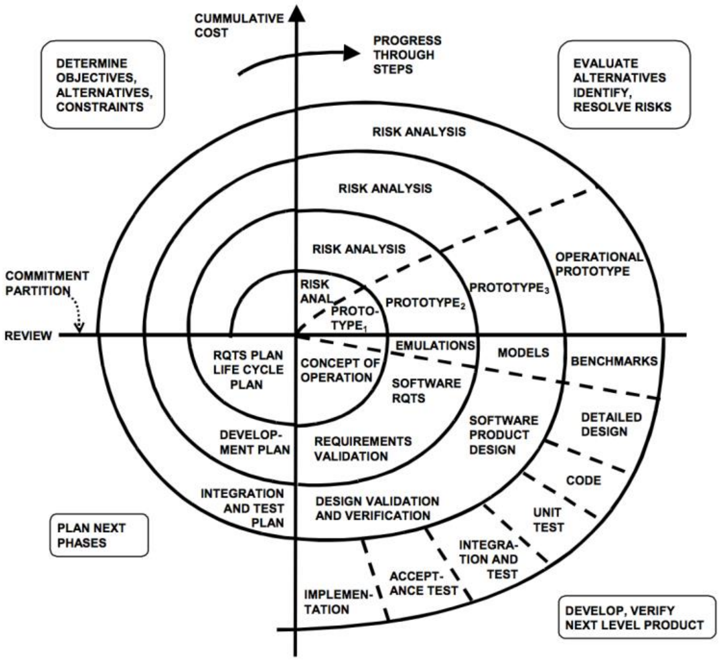


Figure 4. Spiral model
(figure reproduced from [28])

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.

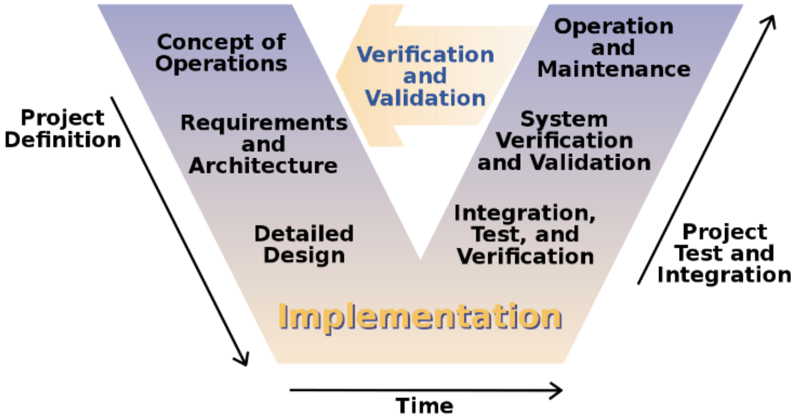


Figure 5. V-Model
(figure reproduced from [29])

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.

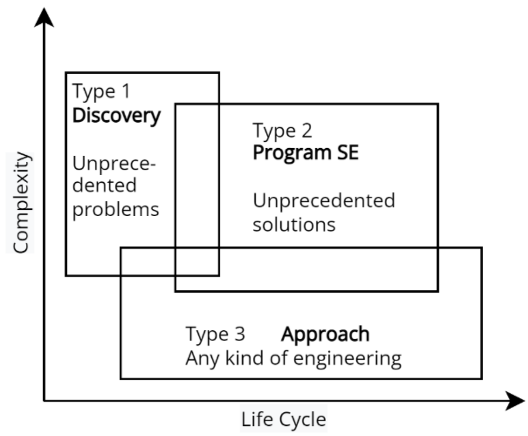


Figure 6. Three types of Systems Engineering implementation (reprinted from [30] with permission)

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 (CMMI) 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	**Requirements engineering and management**
▪ Cost and schedule baselines	▪ Customer requirements specification
▪ Integrated master plan	▪ System requirements specification
▪ Integrated master schedule	▪ Use cases
▪ Integrated product teams	▪ Requirements acceptance criteria
▪ Technical approach	▪ Requirements allocations
▪ SE master plan	▪ Requirements approval process
▪ SE master schedule	▪ Requirements impact assessments
▪ SE processes	▪ Requirements management system
▪ Work breakdown structure	**Project integration**
Project monitoring and control	▪ Product integration process
▪ Earned Value Analysis (EVA)	**Test and verification**
▪ Peer review plan	▪ Verification procedure
▪ Review of action items	▪ Verification entry and exit criteria
▪ Review of issues	▪ Verification criteria
▪ Review process	**Validation**
▪ Review of selection criteria	▪ Validation procedure
▪ SE tracking records	▪ Validation criteria
Risk management	**Configuration management**
▪ Risk list	▪ Configuration baselines
▪ Risk mitigation plans	▪ Configuration item list
▪ Risk mitigation status	▪ Baseline archives
Architecting	▪ Baseline audit records
▪ Concept of operations	▪ Change control board
▪ Product architecture	**Trade-off studies**
▪ Interface descriptions	▪ Alternate solutions
▪ Interface control documents	▪ 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 Quality Management System (QMS) requirements (ISO 9001), QMS requirements for Medical Devices (ISO 13485), QMS requirements for Aviation, Space, and Defence (AS9100), and QMS requirements for the Food and Drug Association (FDA) (21 CFR Part 820).

Table 3 lists the formally established and controlled procedures that apply to (parts of) the general design and development process. Note that more procedures exist; those that do not belong to design and development are omitted. Procedures are either controlled globally by corporate (prefix BE-) or by Benchmark Almelo (prefix AN-).

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.

Table 3. List of corporate and Almelo procedures

Document nr.	Rev.	Title
BE-11003	H	Engineering Design Control Methodology
BE-11004	J	Design and Development Plan Procedure
BE-11005	E	Design History File Procedure
BE-11006	G	Design Input Procedure
BE-11007	G	Design Change Procedure
BE-11008	F	Medical Risk Management Procedure
BE-11009	G	Design Output Procedure
BE-11010	F	Design Review Procedure
BE-11011	E	Design Verification Procedure
BE-11012	F	Design Validation Procedure
BE-11013	F	Design Transfer Procedure
BE-11014	E	Software Engineering Process
BE-11015	B	Medical Usability Engineering Procedure
BE-11018	E	Design Failure Mode and Effect Analysis
AN-11002	5.0	Phase Completion Review process
AN-11007	2.0	Engineering Project Management Procedure
AN-11015	3.0	Mechanical Engineering Process
AN-11020	4.0	Electronics Engineering Process
AN-23003	16.0	Documents and Records Control Procedure
AN-24100	8.0	Engineering Change Process
AN-04600	1.0	New Product Introduction Process

4.1. Engineering Design Control Methodology

The project life cycle stages as defined by Benchmark are documented in BE-11003 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. It integrates the various design control procedures listed in Table 3.

For each project, a technical PM is appointed who is responsible for managing the project to completion through all the life cycle stages. It is the responsibility of the PM to ensure compliance to applicable standards and procedures. Furthermore, the PM directs all decisions and actions of the team, and they represent Benchmark to the customer regarding all aspects of the project.

The PM ensures timely execution of the project according to the approved Design and Development Plan (DDP) and ensures that the customer requirements are represented in the product. To track open issues and to allow traceability of issues back to their origin, the PM maintains an action register for the project through all phases. They are also responsible for creating and maintaining the Design History File (DHF) as described in BE-11005 (see also Section 4.4.4).

Figure 7 gives a graphical representation of the project phases as defined in BE-11003 and Table 4 presents an overview of general activities performed in each phase. Some projects skip the first phase (Technology Development). This is a consideration to be made based on the technological risk of the project, and to what extent the customer has already defined product requirements.

The subsections after the table discuss for each phase the goals, and in- and outputs in more detail.

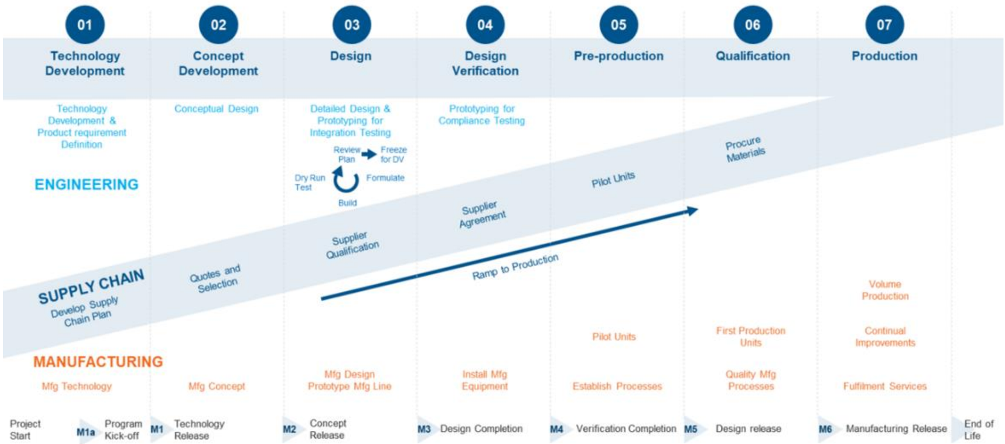


Figure 7. Diagram showing Benchmarks seven project phases (reproduced from BE-11003)

Note the diagonal band dividing Engineering and Manufacturing: as the project progresses through the stages, the involvement of the Engineering department dwindles and makes way for Manufacturing. Supply Chain, represented as the diagonal itself, is involved ubiquitously in all phases.

Table 4. 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 Product Requirements Specification (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 Design FMEA (DFMEA) Usability assessment Create Design Verification Plan (DVP) Create conceptual designs

3	Design	Finalize PRS, Mechanical Requirements Specification (MRS), Electronic Requirements Specification (ERS), Software Requirements Specification (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

Note that the BE-11003 Engineering Design Control Methodology does not include the support or end-of-life stages of the product. Separate procedures exist for sustaining engineering, however these are out of scope.

4.1.1. Kick-off

Depending on the client, design projects may start at any phase. However, projects are always started with a kick-off and a subsequent project proposal. The main goal of the kick-off is to establish the initial scope of the program, to create and review the DDP, and to assemble the project team. Often such a kick-off involves a customer visit with part of the team to carry out preliminary requirements elicitation to determine the project scope and to build mutual trust between Benchmark and the customer.

The key inputs, activities, and outputs of this phase are documented in the Input-Process-Output (IPO) diagram in Figure 8.

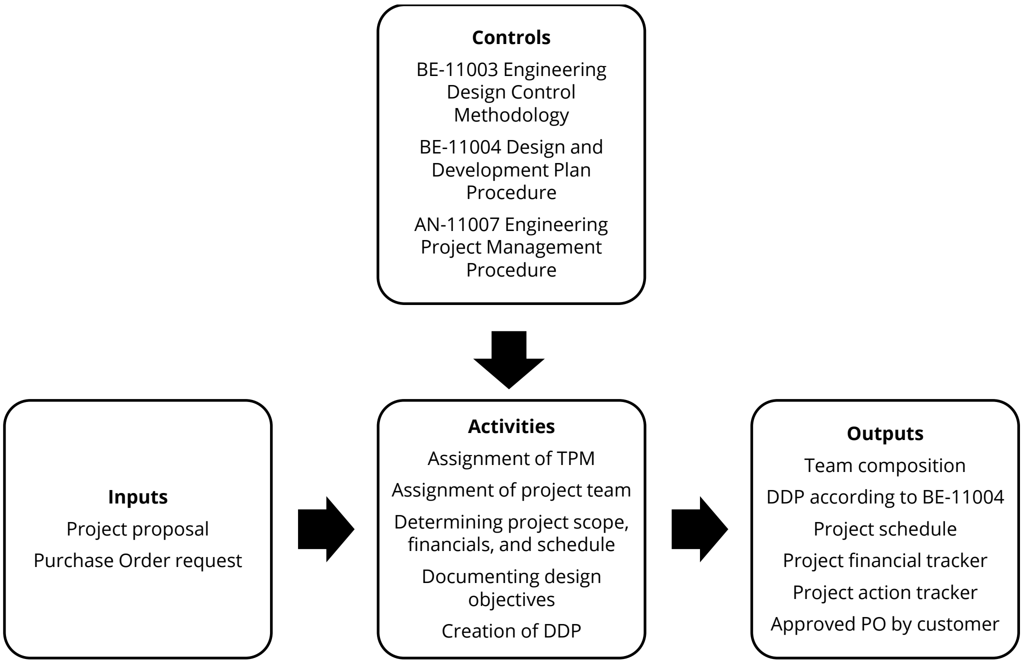


Figure 8. IPO diagram of the kick-off phase

4.1.2. Phase 1. Technology Development

The Technology Development phase represents a 'discovery' phase, to define the product and project before investing significant resources. Taken verbatim from BE-11003, "the goal is to demonstrate basic product, production process, and design feasibility". In other words, this phase is where the TRL is assessed, proofs-of-concept are built, and the team performs preliminary feasibility analysis. This phase includes various risk reduction activities like go/no-go exercises, critical function deployment, and continued schedule and budget assessment. This phase's IPO diagram is shown in Figure 9.

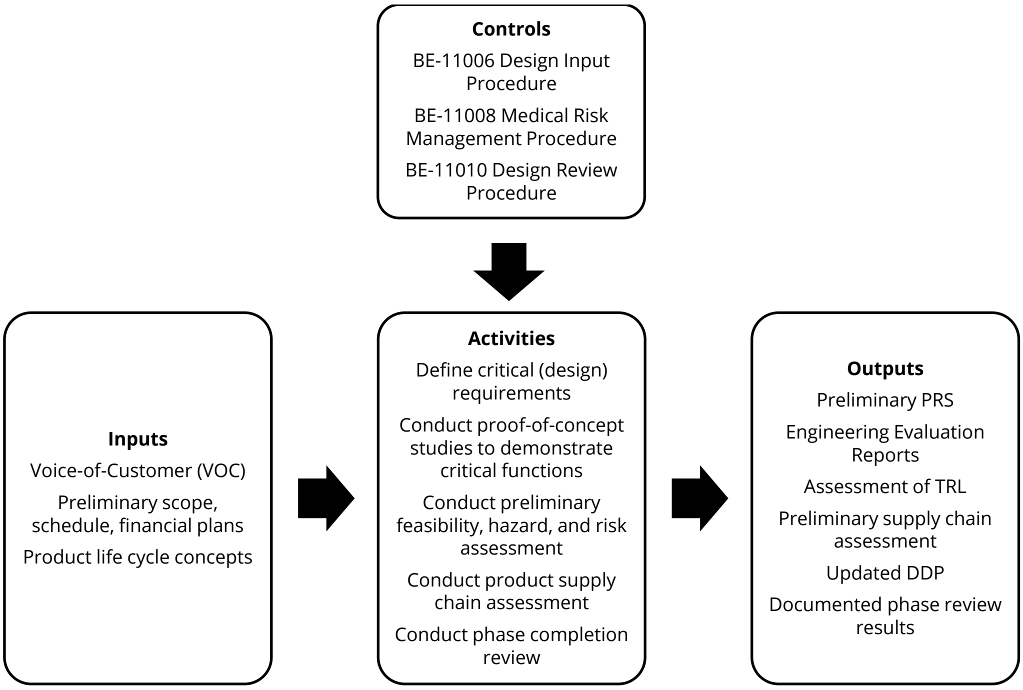


Figure 9. IPO diagram of the Technology Development phase

For phase completion, the preliminary PRS and the DDP must be approved by Benchmark and the customer. To demonstrate completion of these items, the following reviews are typically held:

- **Proof of Technology Review:** Joint internal and external review to decide if the proof-of-concept studies substantially demonstrate the operational concept to meet the PRS.
- **Technology Development Phase Completion Review:** Joint internal and external review to show the deliverables of the phase are successfully completed. Assessment of TRL, program plan, PRS, risk assessment, feasibility/hazard assessment, and proposed concepts for a product.
- **Approval of Milestone Completion:** Management review of all phase deliverables, financials and schedule assessment, and final go/no-go for progressing to the next phase.

4.1.3. Phase 2. Concept Development

In this phase, design/product/system inputs are gathered and synthesized into a conceptual design. Preliminary inputs are gathered from the Technology Development phase, if it was carried out. Activities include the creation of a DHF, definition (or update) of all design inputs, requirements, creation (or update) of Risk Management Plan, initial DFMEA, creation and evaluation of conceptual designs, and initial design verification. The IPO diagram for this phase is shown in Figure 10.

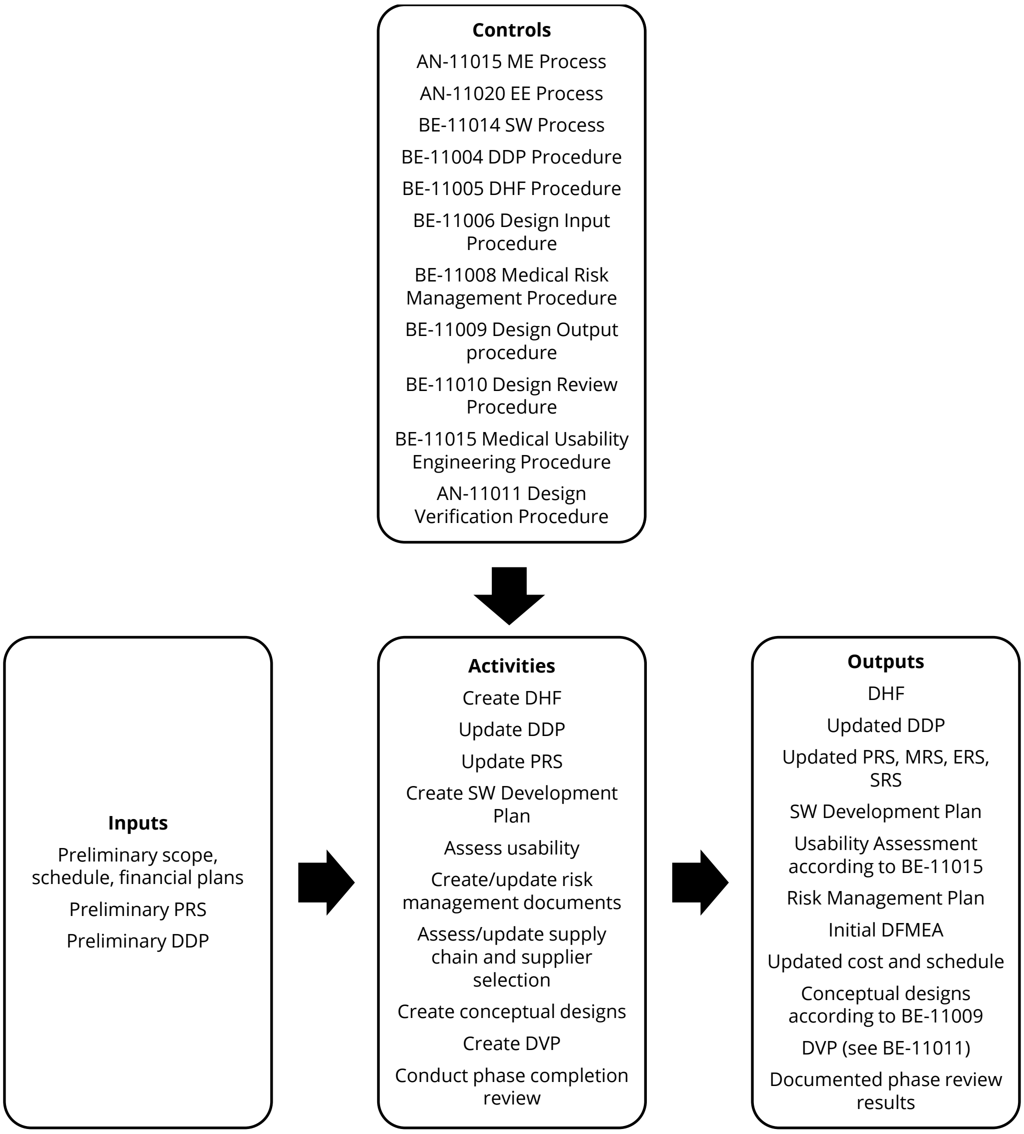


Figure 10. IPO diagram of the Concept Development phase

To demonstrate completion of the required deliverables (as defined in the DDP), and readiness to proceed to the next phase, the following reviews are typically held:

- **Electrical Concept Review:** Joint internal and external review to verify the electrical design, requirements, and tests have fully and unambiguously been defined, have reasonable potential to be successfully verified, and are suitable for generating schematics, Printed Circuit Board (PCB) layouts, and prints.
- **Industrial Design Concept Review:** Joint internal and external review to verify the Industrial Design and user interactions match the expectation of the customer and

have reasonable potential to be successfully verified, and are suitable for detailed design activities.

- **Mechanical Concept Review:** Joint internal and external review to verify the mechanical design, requirements, and tests have fully and unambiguously been defined, have reasonable potential to be successfully verified, and are suitable for generating CAD visuals, drawings, schematics.
- **Software Concept Review:** Joint internal and external review to verify the software design, requirements, and tests have fully and unambiguously been defined, and are suitable for input to the software design phase, according to Software Development Plan (see BE-11014).
- **Concept Development Phase Completion Review:** Joint internal and external review to show the deliverables of the phase are successfully completed. Assessment of program plan, DDP, PRS, system design definition readiness to enter Design Phase, draft design verification and validation strategy, risk assessment, preliminary bill-of-materials (BOM) review, and proposed material and supplier selections.
- **Approval of Milestone Completion:** Management review of all phase deliverables, financials and schedule assessment, and final go/no-go for progressing to the next phase.

4.1.4. Phase 3. Design

The Design Phase is an iterative phase where the product is defined to increasingly more detail. The phase includes creation of documentation, engineering drawings, schematics, SW source code, BOMs, prototypes, and documentation with subsequent reviews and approval. Figure 12 presents the IPO diagram for this phase.

Multiple rounds of prototyping typically occur in this phase, each focussed on covering the design with increasing comprehensiveness and coverage of the requirements specification. The prototyping process follows the following stages:

1. **Review & plan:** Planning/reviewing the current standing.
2. **Formulate:** Formulating requirements, components, functions, test procedures.
3. **Build:** Building prototypes adhering to specifications defined in Formulate stage.
4. **Test:** Testing of prototypes according to the procedure defined in the Formulate stage.
5. **Design Freeze:** The completed design is frozen for entry to the next stage.

Each of these stages have their own phase completion requirements, forming a cycle within the overarching Design Phase (see Figure 11). Once the prototype has demonstrated sufficient design completeness, the design enters the Design Freeze stage. In this stage, the design documentation is to be updated and finalized to progress into the Design Verification phase. Any changes made after the Design Freeze stage must comply with the formal Design Change Procedure defined in BE-11007.

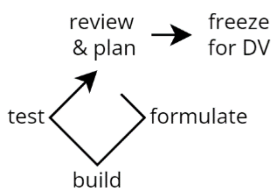


Figure 11. Prototyping cycle within the Design phase

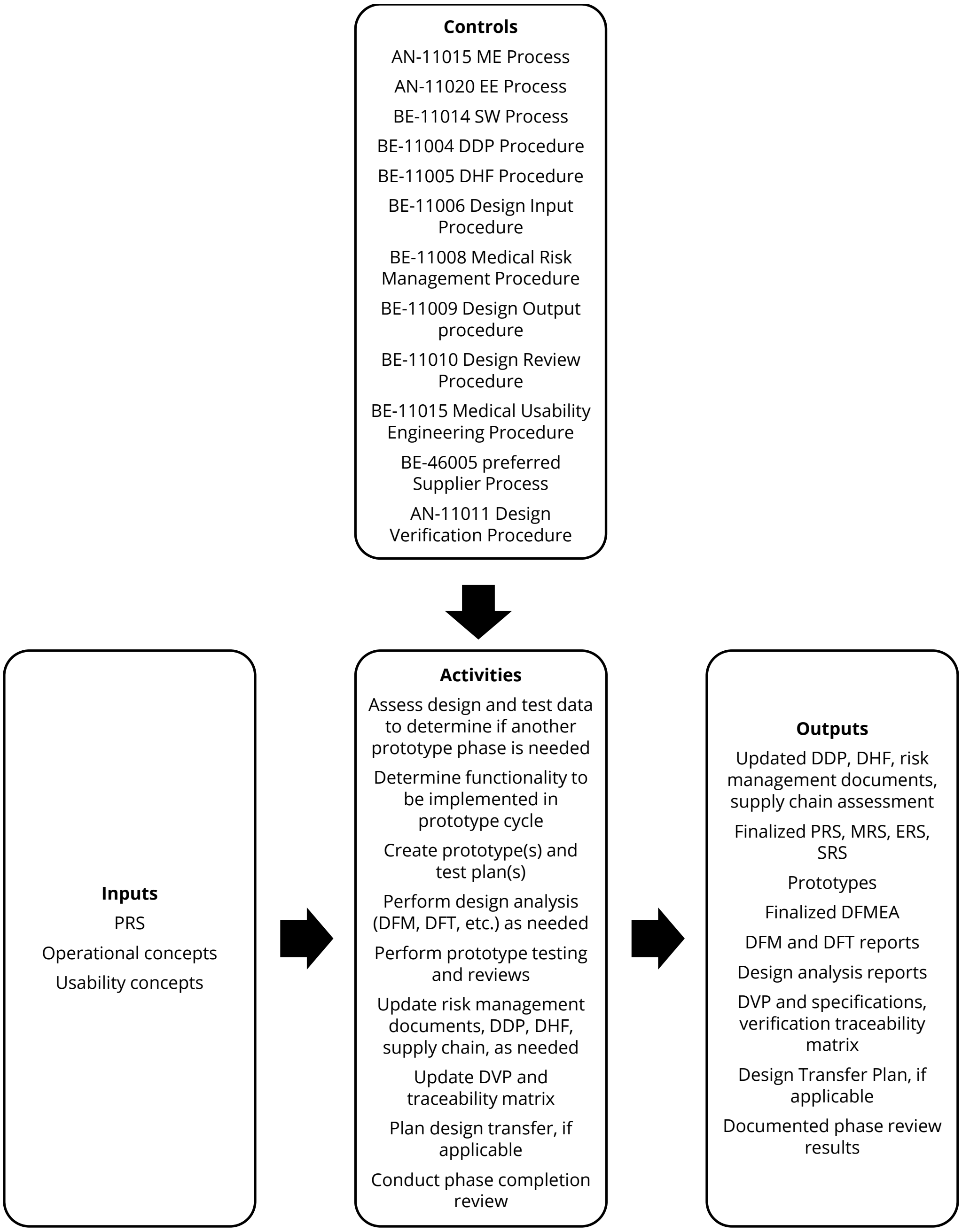


Figure 12. IPO diagram of the Design phase

As part of this phase completion reviews, the following reviews are typically held:

- **Electrical Design Review:** Joint internal and external review to check if the design inputs are defined properly and conflicting requirements (trade-offs), if applicable, have been considered.
- **Mechanical Design Review:** Joint internal and external review to check if the design inputs are defined properly and conflicting requirements (trade-offs), if applicable, have been considered.
- **Software Design Review:** Joint internal and external review to check if the design inputs are defined properly and conflicting requirements (trade-offs), if applicable, have been considered. Decide if software is completed, also including documentation, testing, and control.

- **Design Development Phase Completion Review:** Joint internal and external review to show the cumulative changes since last design review have a reasonable potential to be successfully verified and validated in the next phase. Assessment of program plan, PRD, DVP plan, software reviews, risk assessment, design input/output traceability matrix, technical design deliverables, validation of PRD test methods.
- **Approval of Milestone Completion:** Management review of all phase deliverables, financials and schedule assessment, and final go/no-go for progressing to the next phase.

4.1.5. Phase 4. Design Verification

In the Design Verification phase, a verification unit of the system is built and evaluated according to the Verification Test Plans that were developed in the previous phase. Verification testing is done according to the BE-11011 Design Verification Procedure.

For systems expected to be transferred to Benchmark in-house manufacturing, design transfer activities continue in this phase. This includes procurement of tooling and components for production.

Programs that have design transfer to an external manufacturer end at this phase.

Figure 13 presents the IPO diagram for this phase.

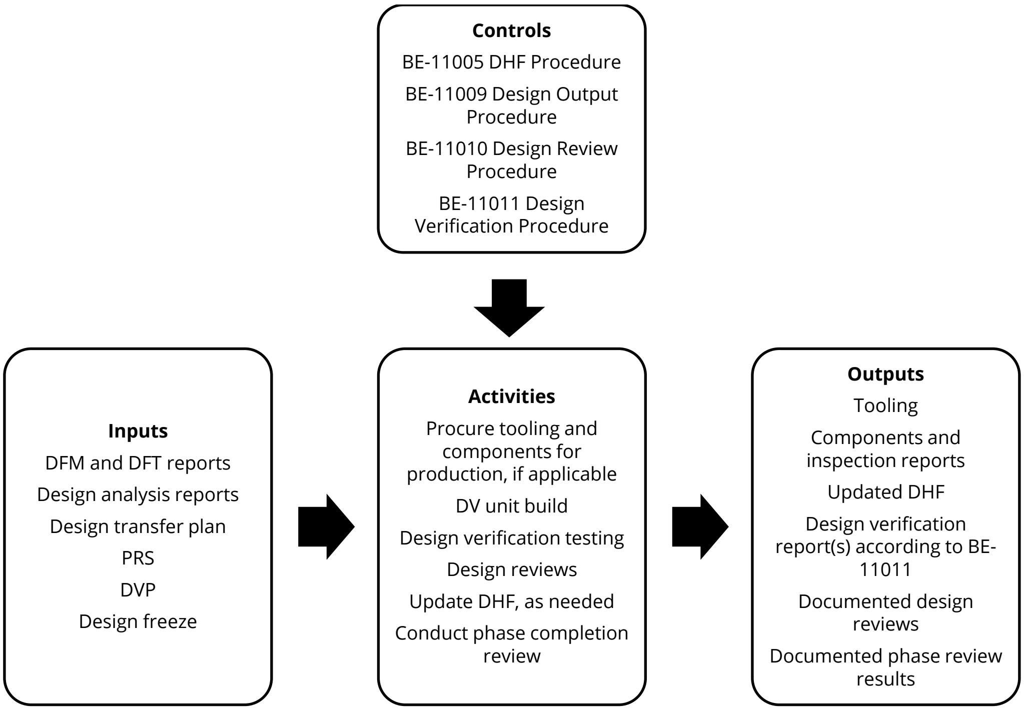


Figure 13. IPO diagram of the Design Verification phase

The following reviews are typically passed to transition to the next phase:

- **Design Transfer Review:** Internal review to verify that the design transfer requirements set out in the Design Transfer Plan have been sufficiently met.
- **Design Verification Phase Completion Review:** Joint internal and external review to verify that the design is correctly translated into production specifications. Assessment of PRS vs. Design Verification Testing (DVT) data, review of DVT report(s),

internal manufacturability review, design and process validation specification and plans.

- **Approval of Milestone Completion:** Management review of all phase deliverables, financials and schedule assessment, and final go/no-go for progressing to the next phase.

4.1.6. Phase 5. Pre-Production

During this phase, Manufacturing will build pilot devices using controlled manufacturing processes. Lessons learnt from the pilot build are used to refine the manufacturing process and Design Engineering performs design verification on the pilot devices according to the DVT plan to ensure that the system performance meets all requirements.

Depending on the product, validation testing may take place during this phase or the Qualification phase. Typically the customer will perform validation testing, due to required specific knowledge or a specific user environment. Design validation activities shall be in accordance with BE-11012 Design Validation procedure.

Next, typically the system risk assessment is revisited and updated after the pilot unit(s) have been built and evaluated. Only minor changes to the system are possible from now on, and only via the BE-11007 Design Change Procedure. Any changes are recorded in the DHF. In the case of significant changes, another round of pilot production may be necessary.

The IPO diagram of this phase is shown in Figure 14.

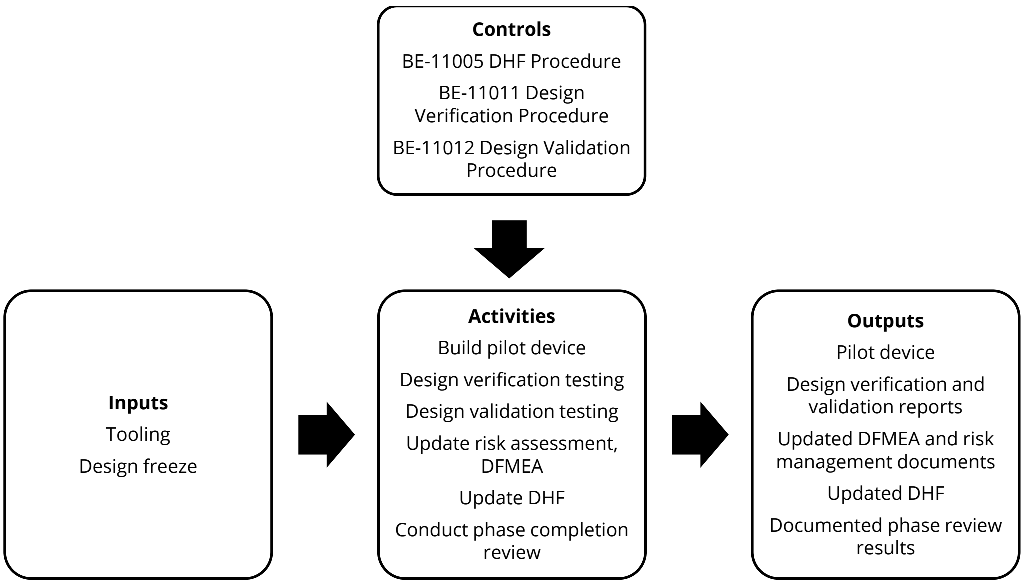


Figure 14. IPO diagram of the Pre-Production phase

The reviews held to finalize this phase are:

- **Pre-Production Phase Completion Review:** Joint internal and external review to verify that the design is correctly translated into production specifications. Assessment of PRS vs. DVT data, review of DVT report(s), internal manufacturability review, design and process validation specification and plans.
- **Approval of Milestone Completion:** Management review of all phase deliverables, financials and schedule assessment, and final go/no-go for progressing to the next phase.

4.1.7. Phase 6. Qualification

In this phase, Manufacturing will build qualification units of the device using released manufacturing processes. All parts of the qualification build will have passed incoming inspection. The qualification units will be verified and validated, and tested according to BE-11011 Design Verification Procedure, BE-11012 Design Validation Procedure, and BE-20005 Global Validation Policy. When applicable, external agency testing will be performed on the qualification devices for CE marking and other accreditations.

All manufacturing documentation shall be controlled in Benchmark' document management system (DMS) (Oracle Agile) system after qualification and validation testing is completed.

During this phase, the primary customer interface role will transfer from the PM to the Benchmark Account Manager, however the PM will still coordinate program oversight until it is clear the transition to manufacturing has been successfully completed.

Figure 15 shows the IPO diagram for this phase.

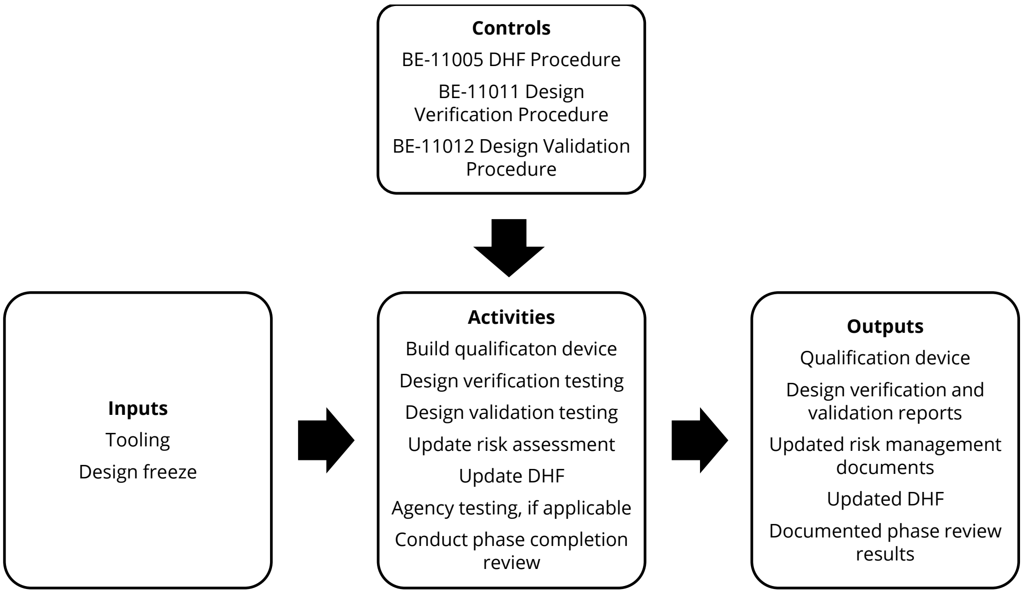


Figure 15. IPO diagram of the Qualification phase

Similar to the previous phase, the reviews typically held to finalize this phase are:

- **Qualification Phase Completion Review:** Joint internal and external review to verify that the design is correctly translated into production specifications. Assessment of PRS vs. DVT data, review of DVT report(s), manufacturability review.
- **Approval of Milestone Completion:** Management review of all phase deliverables, financials and schedule assessment, and final go/no-go for progressing to the next phase.

4.1.8. Phase 7. Production

This phase concerns the production of the final product according to controlled and verified production processes. The production phase must be preceded by a design transfer. Any changes to released documents after the design transfer are handled through a manufacturing engineering change order.

As described in BE-11003, for medical products, any changes after design transer are not recorded in the DHF—it stays static [36]. Instead, changes are recorded in the Device Master Record, as recorded in BE-11005 Design History File Procedure.

Figure 16 shows the IPO diagram for this phase.

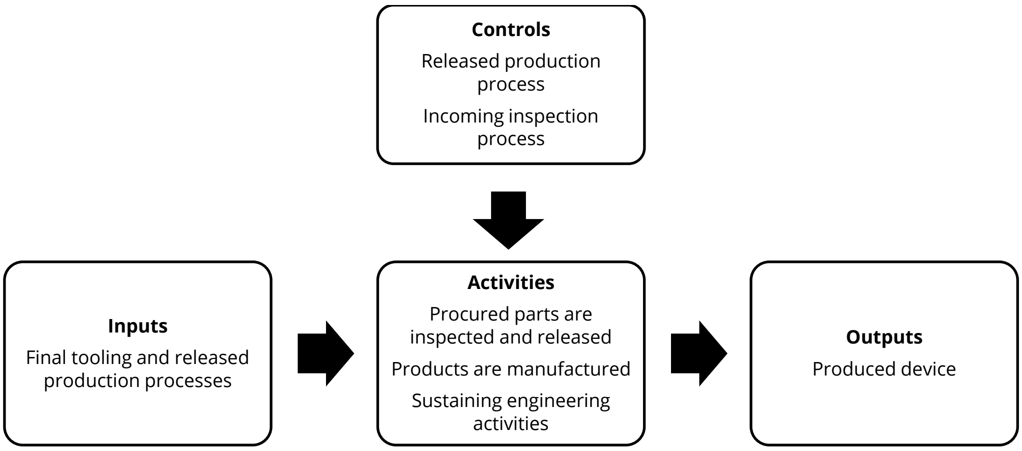


Figure 16. IPO diagram of the Production phase

4.2. Discipline processes

Next to the overarching BE-11003 design process, a step-by-step engineering procedure is defined for the EE, ME, and SW disciplines [37] [38] [39]. These procedures are not inherently linked to BE-11003, but instead define the specific discipline processes in the overall development process.

The procedures are presented in the form of process flowcharts with defined inputs and outputs. From Requirements Definition to TDP Creation, these cover the various activities and deliverables needed for successful design, verification and validation, and acceptance by the customer. They also include references between the engineering disciplines (e.g. SE[2]/HW/SE/ME/Test).

Each engineering process has different activities and deliverables, and focuses on different aspects of the product development process, but all work together to create a complete system. Inputs and constraints from each process are taken into account in the others. Despite the difference in lay-out and notation, 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 Technical Product Documentation (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.

[2] although a process for this does not exist, references are sometimes made to it

The discussed processes are generally followed well in practice. This is because the processes are kept general and non-specific, where possible. Some key activities refer to rigid templates, but for many the interpretation is left to the reader. This means that a lot of responsibility is given to the PM, to determine for each project what deliverables to include and what templates to use. It is important that the SE Process is defined in the same manner: it should describe what to do, but not explicitly how to do it. However, more specific control may be exerted by creating specialized templates.

The variation in the lengths of the processes is significant: the SW Process encompasses 54 pages, the EE Process 22 pages, and the ME Process only 8 pages. However, in essence the three processes allow the V-Model to be followed for the entire design process (the system being detailed from system level to discipline level). Considering this enables us to understand better how they are meant to work together. This makes it possible to connect and integrate the three processes into an overarching SE approach, despite the difference in notation and detail, which makes this analysis very important input for Chapter 7.

The next sections discuss the three separate discipline processes in detail.

4.2.1. Electronics Engineering Process

The AN-11020 EE Process document outlines the process for the electronics development of a system. Described are all EE processes with their inputs, outputs, and responsibilities, including links to the other engineering processes.

The process is summarized in Table 5. For the original process diagram see Appendix A.

Table 5. Summary of the EE Process

Requirements Specification In this activity the ERS is created, based on the PRS and Electronics Concept Description (ECD) developed by HW systems engineers during the concept phase of the project. For very small projects this activity may be omitted with the electronics requirements instead included in the PRS.
Detailed Design The Detailed Design activity involves creating detailed schematic diagrams, selecting components, simulating (sub) circuits, and documenting design choices, taking into account input from the ECD, ERS, and constraints from software, mechanical, and test Engineering. As part of this activity a number of documents (Electronics Design Description, Electronics Verification Plan, ESID, TRS, PCB Layout Input) are produced, as well as design data (simulation results, schematics, BOM). This activity is typically repeated multiple times to create prototype A, prototype B, and pilot products.
Detailed Design Review Before the PCB layout activity, the Detailed Design is reviewed by a team including peer electronics engineers, software engineers, test engineers, systems engineers, and the customer. Potentially also included are mechanical engineers, manufacturing engineers, and layout engineers. The reviews are performed with a focus on functionality, electronical performance, safety, testability, and Electromagnetic Compatibility (EMC)/Electrostatic Discharge (ESD) performance.
PCB Layout The detailed PCB Layout is created in this activity, using input from the detailed design activity and the mechanical design process, while also following design-for-manufacturing (DFM) and design-for-testing (DFT) rules, considering EMC/ESD behaviour, and ensuring that electronics requirements such as signal integrity and high voltage clearances are met.
PCB Layout Review Before ordering the PCB, the PCB layout is reviewed by the responsible electronics engineer, manufacturing engineer, mechanical engineer, and a peer PCB layout engineer, potentially also including a test engineer and the customer, with a focus on electronical

performance including EMC/ESD and signal integrity, mechanical interfaces, testability, manufacturability at the PCB manufacturer and assembly, and safety.
Prototype Build During this phase, actual prototypes are built with parts ordering and board assembly typically handled by PCB assembly. A prototype build report containing remarks from operations about the design, often related to manufacturability or testability, is created.
Design Verification This first verification phase is the first level of testing with the goal of verifying that the electronics design and other product design items such as Field-Programmable Gate Array/SW meet specified requirements. If all requirements are met, the next phase can proceed, but if not, the design may be changed or the PRS or ERS may be adjusted for the unmet requirements, potentially requiring a return to the *Detailed Design* phase.
Product Verification SE is responsible for the Product Verification phase, which is the second level of testing with the goal of verifying that the product meets specified input requirements and legislative requirements as outlined in the PRS, potentially requiring a redesign cycle if all requirements are not met before proceeding to the next phase.
TPD Creation The TPD needed for manufacturing the product is created in this phase, with other disciplines such as mechanical engineering, software engineering, test engineering, and manufacturing engineering responsible for their respective sections.
Design Readiness Review and Design Release The Design Readiness Review is conducted to check the completeness and quality of the TPD before the design can be released for manufacturing, with SE responsible for this process step and signing the TPD release form if all requirements are met.

4.2.2. Mechanical Engineering Process

The AN-11015 ME Process document outlines the process for the mechanical development of a system. Described are all ME processes with their inputs, outputs, and responsibilities, including links to the other engineering processes.

The ME Process is considerably less detailed than the EE Process. Instead of listing all activities with a short description describing the parties involved, it only lists the deliverables that fall under ME responsibility.

The process is summarized in Table 6. For the original process diagram see Appendix B.

Table 6. Summary of the ME Process

Proof of concept In this activity, through proof of concept and principle studies, the concept and PRS are checked to determine whether it is feasible or not. No outputs for this activity have been defined.
Requirements Specification The product level requirements and testing documents (PRS, Product Test Plan [PTP]) are translated into their mechanical-level counterparts (MRS, Mechanical Requirements Traceability Matrix, Mechanical Test Plan [MTP]), thereby making it possible to verify the mechanical implementation of the system.
Design This activity involves creating detailed design data, artwork, inspection sheets, selecting components, and documenting design choices and calculations. Design decisions are documented in the Mechanical Design Description and User Interface Design Description and design data is created (simulation results, schematics, BOM). This activity is typically repeated multiple times to create prototype A, prototype B, and pilot products.
Design Review The mechanical design is reviewed by the responsible mechanical engineer, manufacturing engineer, and Industrial Design and ME Manager, potentially also including a test engineer and the customer. The process also lists a DFMEA as potential output here.
Parts Verification In this activity, before prototyping, the incoming parts are inspected by the responsible mechanical engineer/designer. Inspection results are documented in an IRS.
Prototyping During this phase, actual prototypes are built with parts ordering. Further part inspections are documented in First Article Inspection reports. A prototype build report is created.
Design Verification In this activity the MTP is executed to verify the specifications stated in the MRS of the product. The results are documented in an Mechanical Test Results document, which is similar to the MTP but also includes a pass/fail conclusion and any remarks.
Design Release The Design Readiness Review is conducted to check the completeness and quality of the TPD before the design can be released for manufacturing, with the SE function responsible for this process step and signing the TPD release form if all requirements are met.

4.2.3. Software Process

The BE-11014 SW development process is extensive. The SW Process is structured differently than the EE and ME processes. Instead of structuring the responsibilities in a linear process flow diagram, the SW Process makes explicit that it follows the V-Model. It is also the only process of the three to document its own risk management, change management, and configuration management processes. However, it must be noted that the process is outdated and a new SW Process is currently under development.

The V-Model on which the SW process is based is shown in Figure 17.

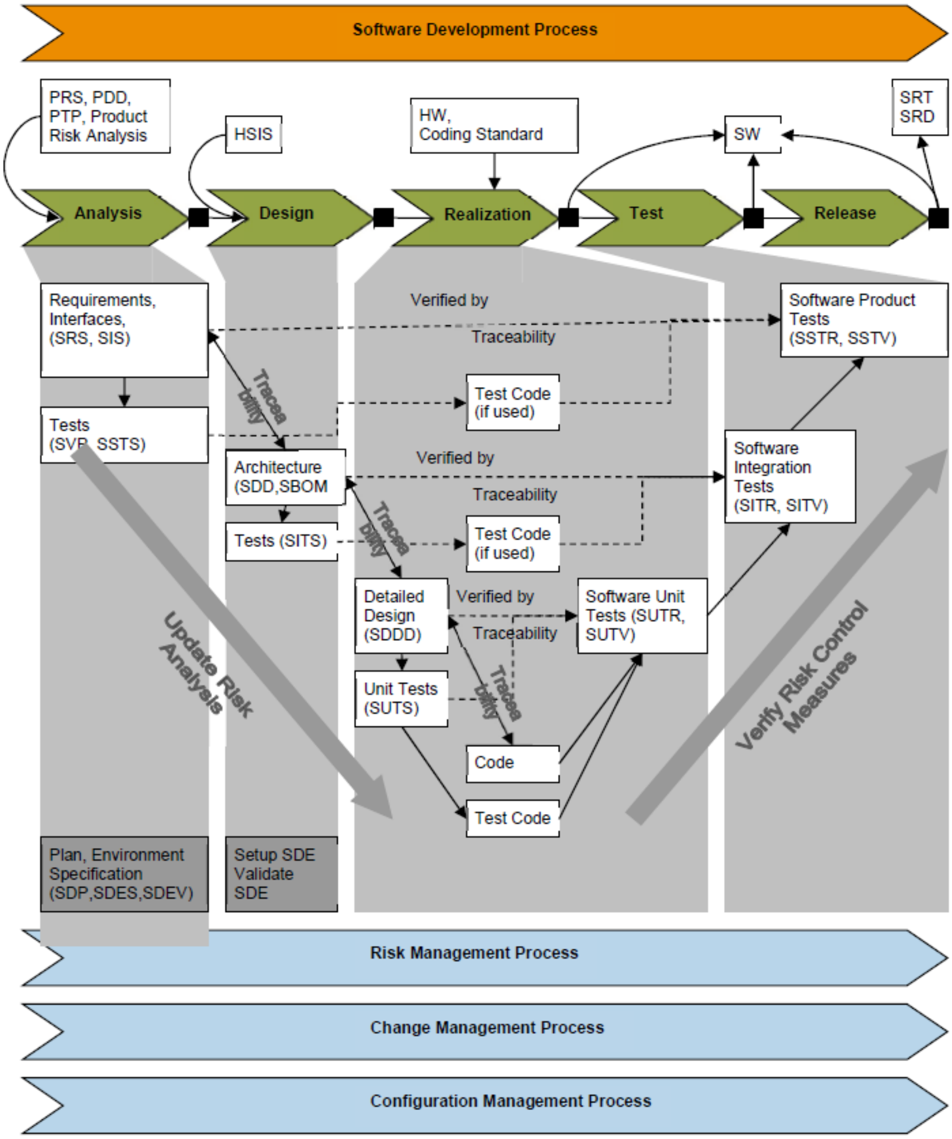


Figure 17. SW Development Process V-Model

In the left part of the SW V-Model, the PRS and Product Design Description (PDD) are broken down into SRS as part of an analysis activity, Software Design Description as part of a design activity, and Software Detailed Design Description and code as part of the realization activity. The right side of the V follows integration, as part of the test activity.

The activities are on the x-axis of the V-model and the level-of-detail of software development is on the y-axis of the V-model. Generally, the V-model is progressed through multiple times in an iterative manner. The defined activities are listed in Table 7.

Table 7. Summary of the SW Process activities

Analysis The purpose of this activity is to gather, communicate, and document the software requirements and tests (SRS, Software Verification Plan), and to provide feedback and support for further development of the inputs from the product level, while also adjusting the software deliverables according to any relevant changes at the product level.
Design This activity involves designing and documenting the software, taking input from the analysis activity, and producing outputs that serve as input for the realization activity. The software design will be documented for all devices requiring software, and the architecture, boundary interfaces, and hardware interfaces are defined during this activity, while also addressing any necessary changes to approved product documentation.
Realization During this activity, detailed designs are made as needed and the software is implemented and unit tested as required, taking input from the design activity, and producing outputs that serve as input for the test activity. The realization activity must also include unit verification for certain software items and may require updates to approved product and software documentation, with the option to automate unit verification to verify the implementation of the detailed design for small software items.
Test The integration and test activity involves integrating software items into larger ones, executing and documenting verification specifications for integration and system levels, and potentially modifying software due to issues discovered during testing. Regression testing is performed to verify that changes to the software did not negatively impact functionality, reliability, or performance or introduce new defects. This activity may also require updates to approved product and software documentation and may involve integration level testing for certain software products.
Release This activity has little documentation but involves delivering verified software and documentation.

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. Documents are released and revision controlled either in the Benchmark DMS (Oracle Agile), or a DMS prescribed by the customer.

A Change Control Board (CCB) evaluates all proposed changes, to ensure there is no adverse impact on the conformity to requirements, based on the following key questions:

- How does the change affect function, performance, usability, safety, or applicable regulatory requirements?
- Does the change require formal design review(s)?
- Does the change require reverification or revalidation?
- Does the change affect risk management?
- Does the change affect manufacturing testing or processes and will it require manufacturing requalification and/or revalidation?
- How does the change impact constituent parts?
- How does the change impact products in progress and already delivered?

Description of the change, its review, and evaluation relative to these questions are recorded in a special format (BEF-11003 Engineering Change Evaluation Form). The change shall also be review approved by the customer when it affects customer requirements or when it is required by contract.

4.4. Noteworthy project deliverables and their contents

This section discusses several important project deliverables that are created and maintained during the design and development process. These include the DDP, Design Input documents, Design Review reports, and the DHF.

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.

The next sections discuss these deliverables in detail.

4.4.1. Design and Development Plan

Pursuant to the requirements of ISO 9001, ISO 13485, AS9100, and 21 CFR Part 820, a DDP is created by the PM in the early stages of a project. For each project, the DDP specifies the design and development activities that permit verification that the design meets the requirements.

The DDP needs to be approved and signed off by the PM, customer, and for especially complex or medical class projects, also the lead engineer(s) and quality assurance. The DDP is maintained and updated through the project phases, and a controlled document is established at every phase-completion review. For medical projects, the controlled DDP is managed in Benchmark's Oracle Agile DMS system, or in a customer imposed DMS system. In any case, strict document change control is in place.

Figure 18 on the next page summarizes the contents of a DDP of a Class III[3] medical project.

4.4.2. Design Input documents

This project deliverable concerns mainly requirements and inputs from the customer. Pursuant to ISO9001 and AS9100, Benchmark shall consider functional and performance requirements, statutory and regulatory requirements, standards and codes of practice, information derived from previous similar design and development activities, potential consequences of failure due to the nature of the product, and when applicable, the possible consequences of obsolescence (e.g. materials, components, equipment, etc.). A list of example design input documents is shown in Figure 19 on the next page.

Design inputs are gathered at the start of a project, as well as during, and are recorded in the applicable specification documents. Part of these documents will originate from Benchmark and part from the customer. All design input documents are part of the maintained DHF.

Design input documents shall be reviewed and approved prior to the phase completion in which they were created. The review and approval panel is defined for each design input document in the DDP. It is important that these documents are structured and kept well, since they inform requirements traceability towards the design outputs and that the product verification plan and test results are compliant, complete, and consistent.

[3] Most severe class of medical product. Only assigned when the product can cause major harm such as severe injuries or death. This is the most complex possible product class for Benchmark, with the most extensive requirements for documentation. Therefore it is chosen as a worst-case example here.

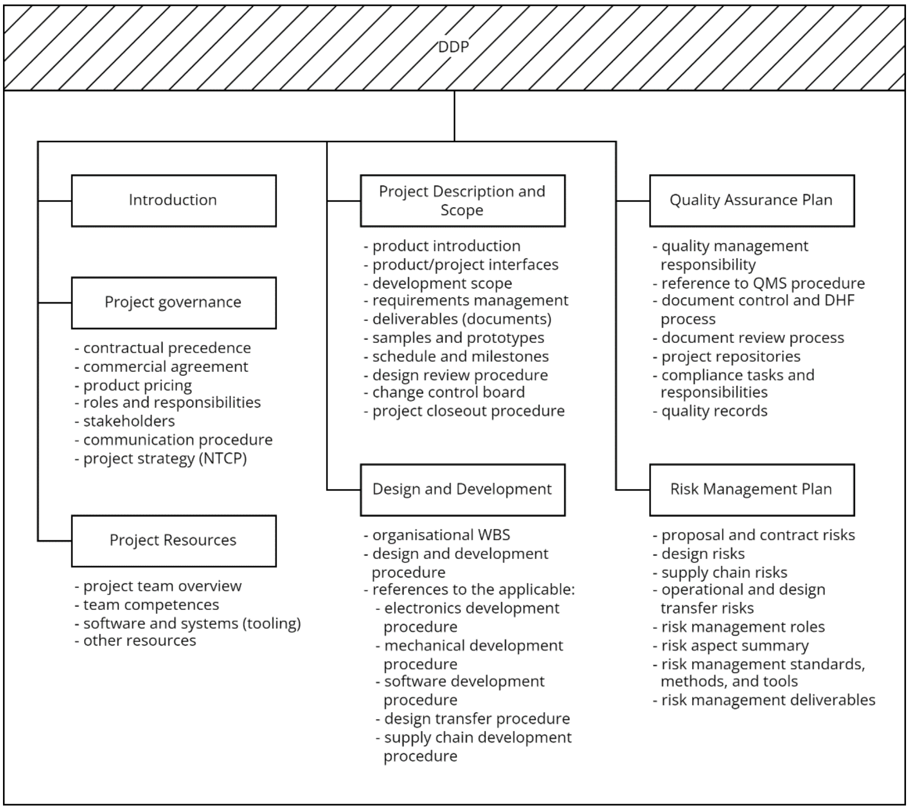


Figure 18. DDP contents

Figure 19. Example design input documents

4.4.3. Design Review reports

Design Reviews ensure a controlled design and development process by evaluating the abilities of the results to meet the requirements. It is evaluated whether necessary actions are taken on problems as identified and determined during the reviews and documented information on these reviews is retained. All of this as ISO 9001, ISO 13485, and AS9100 prescribe.

For each project, the design review procedure is defined in the DDP. Depending on the size and complexity of the project, design reviews may be planned at the end of each project phase, or intermittently at appropriate project milestones. The participants of each review include representatives of functions concerned with the design phase being reviewed, and at least one individual who does not have direct responsibility for the design phase under review.

Design Reviews are documented in a written report and document control is applied accordingly. Open issues identified during review sessions are logged in the project action register and tracked until closure. All design review reports in the end are included in the DHF.

4.4.4. Design History File

The DHF is a repository of documents associated with a project. It contains all design and development documentation, including version history, and contains or references the records necessary to demonstrate that the design was developed in accordance with the approved DDP. Figure 20 below summarizes the contents of the DHF for any project.

The DHF can be organized either by project phase or activity depending on the complexity of the project. It is maintained from the start of the project through to the end by the PM.

It is paramount that the DHF is maintained and organised well. This is because after completion of the project, the DHF is most often transferred to the customer, where it is stored for at least the expected life of the product. Only in exceptional cases is the DHF not transferred and stored in-house. Furthermore, all medical projects are subject to DHF audits by Quality Assurance before production release of the product.

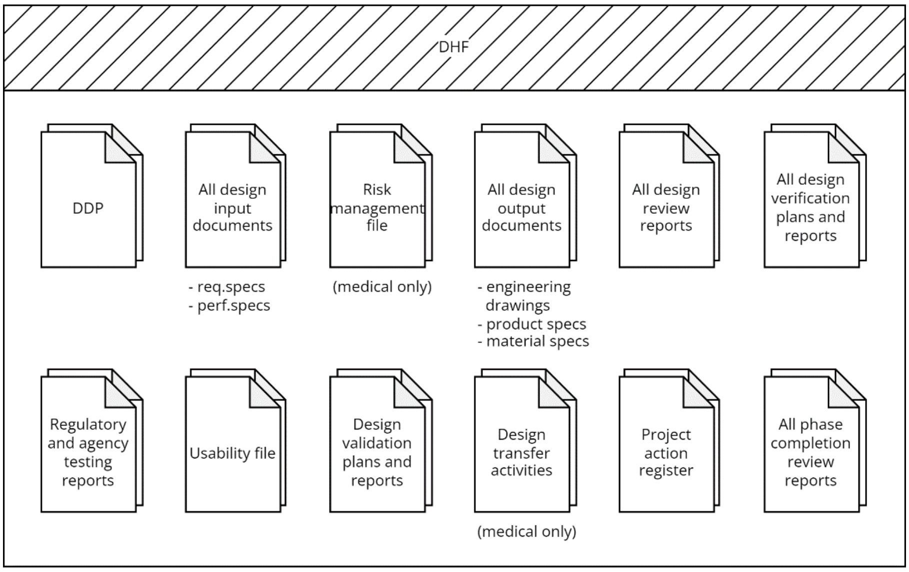


Figure 20. DHF contents

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.

All of the discussed processes and procedures prescribe a way-of-working to ensure compliance to a variety of external standards. These standards include general ISO norms for QMS, but also more specific guidelines for the aerospace and medical industries. A result of this is that the processes follow a well-defined structure. The V-Model, with Waterfall and Spiral structures as intermediaries, is recognizable in many of the processes. Thus, many SE elements already exist. The new SE process should make these elements more explicit.

The discipline processes are the most important input for Chapter 7. They describe the activities that should be performed by the disciplines over the system life cycle. However they are kept general on how exactly to perform the described activities, instead referring to separate templates for more rigid guidance. The SE Process must link the existing three processes together, and can be defined in the same manner: it should describe what to do, and when, but not explicitly how to do it.

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.

The interviews were transcribed where possible (not all interviews could be recorded due to confidentiality) and summarized. The transcripts were coded and theme analysis was performed to identify common patterns by looking for connections and commonalities between the codes.

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 Integrated Master Plan 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 Earned Value Analysis (EVA) baseline updated? Is EVA linked to WBS and IMS?
- How are customer PRs handled?

Four separate PMs 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. Benchmark does not differentiate between junior and senior PM roles, however in some cases a separate project manager is employed below the PM. Despite this, the PM is always heavily involved and within Benchmark generally regarded as 'CEO of their own business'. For this reason, it was decided to interview only PMs[4].

The projects are all from the Design Engineering department, however with a large variety in complexity, size, and sector. Table 8 presents an overview of the projects that were assessed via an interview.

Table 8. Projects assessed via interview

Program manager was interviewed about:	
	Small project for a client in the industrial electronics domain
	Two large projects for clients in the medical domain (medical class II and -III[5])
	Multiple small projects for a long-running client in the industrial electronics domain
	Large project for a client in the avionics domain (DAL-D[6])

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 [40], which was originally a part of the CMMI [33] [41]. 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.

To accommodate a large number of companies of various sizes and industries, the original survey was intentionally kept general. Therefore, to make it suitable for assessment specifically within Benchmark, certain questions were removed and others were rephrased. Despite these changes, the general concept and purpose of the assessment, as well as the scoring process, were maintained. The survey used in this thesis, the scoring procedure, and results are all documented in Appendix C.

The adapted 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

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

[5] Highest and second highest medical device classifications. For devices with high risk that require premarket approval by the FDA and for devices with moderate to high risk that require special controls imposed by the FDA, respectively.

[6] Second lowest Avionics Design Assurance Level D (DAL-D). Failure in this case may cause inconvenience, but not stress, injury, or death.

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.

Additionally, next to the process areas, project complexity and project success are assessed. These additional two measures are very important since they give insight in to what extent SE may have been necessary in the project. For instance, it is not a problem if a highly successful project of low complexity scores low on the capability assessment; apparently more involved SE was not necessary.

The projects assessed via the survey are shown in Table 9.

Table 9. Projects assessed via survey

Program manager filled in survey about:	
	Large project for a client in the industrial electronics domain
	Small project for a client in the consumer electronics domain

5.2. Results

5.2.1. Interview

Only two of interviews could be recorded and transcribed due to confidentiality arrangements. Notes were taken during the interviews that were not recorded, although they will not be published. Nevertheless, important contextual knowledge was gained from these and with the notes it was possible to include the non-transcribed interviews in the thematic analysis process as well. Below follows a summary of the topics discussed. See Appendix D for the interview transcripts.

Summary

The interviews revealed common themes and challenges related to implementing new technology, the importance of good communication and collaboration, and the need for traceability and thorough documentation. The PMs of the largest projects emphasized the need for a systems engineer role that should exhibit a top-down approach to technical management and have a broad understanding of technology.

In one of the interviews, the PM discussed the challenges and approach taken for a new industrial electronics product that was an upgrade of an existing one. The main challenge was implementing a new technology standard. The engineering team at Benchmark lacked experience with this standard, so they hired a third party to handle it. However, communication with the third party was poor, causing delays and increased cost. For this situation, the PM highlighted the importance of good collaboration with clients and other external stakeholders, and that a dedicated systems engineer could have been beneficial.

In a separate interview, the PM discussed the roles and responsibilities of the different team members involved in their portfolio of projects. They mentioned that the existing electronics systems architect role is closest to the systems engineer as defined in this thesis. They indicated the architect is loosely responsible for the requirements and design, but in practice it is a group effort and the responsibility is not clearly defined.

The PMs in the medical, industrial, and avionics domains discussed the importance of traceability and the use of tools such as Word, Excel, and IBM Rational DOORS™ or Atlassian Jira™ for tracking requirements. They emphasized how these tools play a role in project quality and knowledge retention. They indicated it is important to retain knowledge in long-running projects by keeping team members on board and maintaining thorough documentation. The PMs also discussed the challenges of implementing automated testing through these tools and how this compliance testing ensures quality.

The importance of traceability throughout documentation was discussed in all interviews. Being able to trace design decisions back throughout requirements documentation and design descriptions is very important for project success. Furthermore, it was widely agreed that good documentation plays a large role in knowledge-retention. For most projects, the PM was satisfied with the level of document control, however requirements traceability was identified as an improvement area.

The topic of the to-be role of the systems engineer was also discussed. The PM interviewed about a large medical project mentioned a lack of experience with some complex new technologies (e.g. optics and lasers). They emphasized that a systems engineer with great expertise in the early stages of the project could have helped identify this lack of knowledge. Furthermore, the importance of good communication and collaboration with the stakeholders to fill such knowledge gaps came up multiple times, which is a key part of SE.

All PMs agreed that the systems engineer should have a top-down approach to technical management and be broadly knowledgeable about the technology applied in the project. One PM mentioned that the systems engineer should sometimes even 'reel in' the development team, when they may be overly excited to use a new technology and therefore overlook limitations or make unfulfillable promises.

Themes
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.
- **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.

However, due to the small sample size (in the end, only two surveys were fully completed) it is difficult to conclude anything definitively. Furthermore, there was a large spread in the quantified scores between the two surveys, since the projects were of vastly different size and scope. In conclusion, the surveys were not as effective in elucidating improvement areas as expected.

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*.

Table 10 shows how the key takeaways from the interviews and survey integrate into four concrete points for improvement.

Table 10. SE distilled key improvement points

SE assessment key take-aways	**Distilled improvement points:**
Interviews: ▪ Collaboration and communication ▪ Top-down technical management ▪ Clear SE role ▪ Traceability and documentation **Survey:** ▪ Integrated product teams ▪ Risk management ▪ Trade-off studies ▪ Requirements management ▪ System architecting	**Improve…** ▪ Collaboration and communication in Integrated Product teams (IPTs) via top-down technical management. ▪ Traceability and documentation via structured requirements tracking and risk management. ▪ System architecting and trade-off studies by better defining the SE role and its responsibilities. ▪ Project quality via better technical leadership from SE and oversight on Verification and Validation (V&V) activities.

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

The first step for successful implementation is to define a clear vision for how the organization should operate in the future. This vision must be specific, relevant, and aligned with the overall vision of the organization.

A good vision defines how an organization will operate in 2-4 years' time in relation to customers, suppliers, employees, society, and shareholders. It is important that this vision is clearly communicated to all members of the organization, and that everyone is involved in its creation, so that everyone understands their role and how they contribute to the overall vision. Referring to Kotter (1996), any vision should be developed so that it clarifies the direction of change in such a way that the entire organization will support and promote it [42] [43].

Establishing this vision involved 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.

Over the course of several months, the following main points were identified for how SE should be implemented at Benchmark.

- **SE should be concretized and made explicit:** The new SE program should formally define the roles and responsibilities of systems engineers towards other stakeholders, as well as the specific steps that need to be taken during the SE lifecycle. It should be flexible enough to be adapted to different types of projects and technologies, but also have a common framework that is followed.
- **The SE role is almost like a cross-over between a lead engineer and technical manager:** The SE should be responsible for a design on system level. They should ensure proper communication and overview of the technical solution and trade-offs. Together with the team they monitor risks and changes to prevent unforeseen errors. And finally a large part of the SE's responsibilities lie in ensuring proper testing and verification of the solution.
- **A culture of continuous improvement should be established:** SE is an iterative process and it is important to be able to learn from experience and make adjustments as needed. The new process could facilitate this by allowing for regular reviews and retrospectives to identify areas for improvement, and encouraging engineers to share their knowledge and best practices with others.
- **Cross-functional collaboration should be encouraged:** SE is a multidisciplinary field that involves collaboration between different teams and departments, such as R&D, manufacturing, and quality control. The new SE process should encourage and facilitate communication and cooperation among teams.
- **Training and personal development must be valued highly:** In order to build a team of skilled and proficient systems engineers, it will be important to invest in training and development programs. This can be done in-house, or through external training programs.

The final vision combining these points is shown in Box 3 on the next page. Table 11 then shows how the vision traces back to the results of the SE assessment.

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."

A. **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.
B. **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.
C. **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.
D. **They ensure proper testing and verification of the solution:** The systems engineer shall drive system-level acceptance testing and support verification 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.

Table 11. Vision topics traceability

SE assessment key take-aways	Distilled improvement points:	Vision topics
Interviews: ▪ Collaboration and communication ▪ Top-down technical management ▪ Clear SE role ▪ Traceability and documentation **Survey:** ▪ Integrated product teams ▪ Risk management ▪ Trade-off studies ▪ Requirements management ▪ System architecting	**Improve…** ▪ Collaboration and communication in IPTs via top-down technical management. ▪ Traceability and documentation via structured requirements tracking and risk management. ▪ System architecting and trade-off studies by better defining the SE role and its responsibilities. ▪ Project quality via better technical leadership from SE and oversight on V&V activities.	A. SE is responsible on a system level. B. SE ensure proper communication and overview of technical solution and trade-offs. C. SE monitor risks and changes and prevent unforeseen errors. D. SE ensure proper testing and verification of the solution.

6.2. Defining Systems Engineering program tasks

In implementing the vision, the first step is defining what activities or tasks in a program belong to SE. Since SE covers a wide range of activities, the main challenge is identifying the program tasks that relate to the system as a whole.

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. In the early stages of a project, the systems engineer plays a crucial role in solution synthesis. As the project progresses, the systems engineer takes on a leadership role, providing guidance for the final implementation of the design.

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 SE program tasks:

1. **Perform solution synthesis:** Exploring the problem domain and coming up with a solution.
2. **Perform requirements and Technical Performance Measure (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.

Table 12 shows how these defined program tasks trace back to the vision topics defined in Section 6.1.

Table 12. Defined SE program tasks traceability

Vision topics	Defined SE program tasks
A. SE is responsible on a system level.	1. Perform solution synthesis 2. Perform requirements and TPM Analysis 3. Perform functional analysis and allocation 4. Create a Concept Design Description
B. SE ensure proper communication and overview of technical solution and trade-offs.	5. Create a System Architecture Overview 11. Ensure proper documentation
C. SE monitor risks and changes and prevent unforeseen errors.	7. Support Synthesis, Analysis, Evaluation 8. Support Risk Management activities 9. Plan, coordinate, and conduct design review meetings
D. SE ensure proper testing and verification of the solution.	6. Support Design Verification and Validation Plan creation 10. Lead integration and monitor verification activities

Tasks 1-4 tackle topic A from the vision: they concretize the responsibilities of the systems engineer in the early stages of the project. The systems engineer is made explicitly responsible for coming up with a solution direction in the fuzzy front-end (task 1), subsequent requirements and functional analysis (tasks 2 and 3), and for the design of the concept (task 4).

Tasks 4, 5 and 11 are focused on topic B from the vision and support communication and documentation by making the systems engineer responsible for the Concept Description (task 4) and for maintaining a System Architecture Overview (task 5), which will improve knowledge-sharing within a project. Task 11 also contributes as it is a transverse activity throughout the whole project.

Tasks 7-9 address vision topic C by formally embedding the systems engineer into a managerial leadership role. It is important that the systems engineer is involved in the continued synthesis, analysis, and evaluation cycle (task 7) in the design process, as they will be able to advice and correct. This also goes for risk management (task 8) and design review activities (task 9) which are transverse over the whole design cycle.

Lastly, tasks 2, 6 and 10 aim to improve topic D by formalizing the systems engineer's responsibility in verification and validation planning (task 6) and execution (task 10). While at Benchmark product validation (testing if the right product is built) is the responsibility of the customer, evaluating the system (task 10) on the *right* set of requirements (task 2) can be a proxy for validation, thereby still giving the systems engineer a part of the responsibility.

The following sections give more detail into what these tasks exactly entail and the rationale for performing them. These tasks concern multiple activities, for instance task 1 *solution synthesis* includes not only requirements elicitation, but also performing feasibility studies. To visualize the activities performed as part of these SE tasks, IPO diagrams are again presented.

6.2.1. Perform solution synthesis

The SE function shall perform a needs and VOC analysis, understand the system context and problem domain, assess the TRL, conduct feasibility studies, and determine the technical solution for the problem.

These initial *solution synthesis* activities have large impact on the remainder and success of the project[7] as they are necessary to identify what problem to solve and how to solve it. They are performed at system-level and therefore are the responsibility and under leadership of the SE function.

It is important to conduct these analyses thoroughly in the infant stages of the project, to ensure a substantiated confidence that the problem is identified properly and can be solved. Proper consideration here ensures the subsequent development efforts are decisive and effective, and it avoids rework and recalls in later stages. It also allows the team to estimate the project timeline and estimated necessary budget, before doing any design work.

For complex new acquisitions with high technological risk, these activities may be carried out via one or more workshops and meetings with the customer or end-user. It is important to carefully consider the problem space together with the customer and potential users. For

[7] as demonstrated; see e.g. the Fluke Industrial Acoustic Imager (Table 1). For this project many of the early analyses were performed in a large, almost week-long workshop to define and explore the solution space early on in the project and to ensure the whole team was on the same page at the start of the project. This resulted in a very fast development turnaround time of nine months from kick-off to production, despite the product being technically challenging.

lower-risk projects with already clear objectives, a strong interface with the customer's (systems) engineering department is necessary, to internalize a potentially already defined solution space. In any case, good communication with the customer and a strong technical leadership role by the systems engineer is paramount.

Also part of this activity is the creation of multiple models, proofs of concept, or early prototypes. These should substantiate the systems engineers confidence in the chosen technology or solution direction.

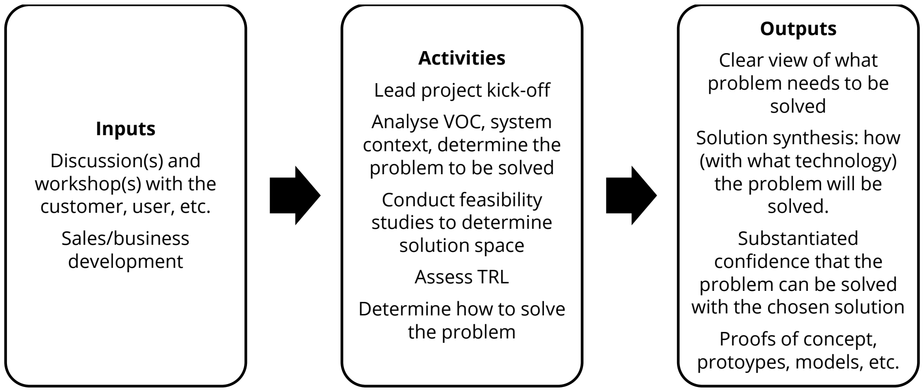


Figure 21. Solution Synthesis IPO diagram

6.2.2. Perform requirements and TPM analysis

The SE function shall conduct requirements analysis to determine top-level critical requirements for the system and analyse and rank TPMs for the system. They shall also keep track of all applicable external standards and regulations that have influence on the design and its requirements.

These activities substantiate the overall definition of system-level requirements and are the basis for the top-down system design. Based on the proposed technical solution and the definition of need, the SE function shall conduct requirements elicitation and analysis. Next to system operational requirements, the SE function should also consider secondary aspects such as maintenance and support, disposability, and end-of-life.

As part of this process it is necessary to establish clear measures of effectiveness for the system, to quantitatively assess to what extent customer expectations are met. To reflect customer priorities these shall be ranked, and in turn will influence the subsequent design process as a set of clear design criteria.

The SE function shall, together with the customer, user, and team, consider all relevant aspects of the design (e.g. producibility, affordability, quality, safety, human factors, feasibility, environment, supportability, compatibility, etc.). Some of these aspects/design considerations may be influenced directly by the design of the system and further development efforts and can therefore be classified as *design-dependent*. Some factors that the system is subject to (e.g. fuel costs, labour rates) are *design-independent*, but should still be noted.

The SE function shall together with the stakeholders define TPMs for the parameters that are design-dependent. These are expected/predicted/estimated values that serve as measures of effectiveness for the design-dependent parameters of the system.

All of this shall culminate in a clear list of *design criteria*: customer specified or negotiated target values for TPMs. These design criteria offer a way of measuring project success in a quantitative way, and are therefore important input for the remaining engineering efforts in the project. Because these considerations happen on a system-wide level it falls under responsibility and under clear leadership of the SE function.

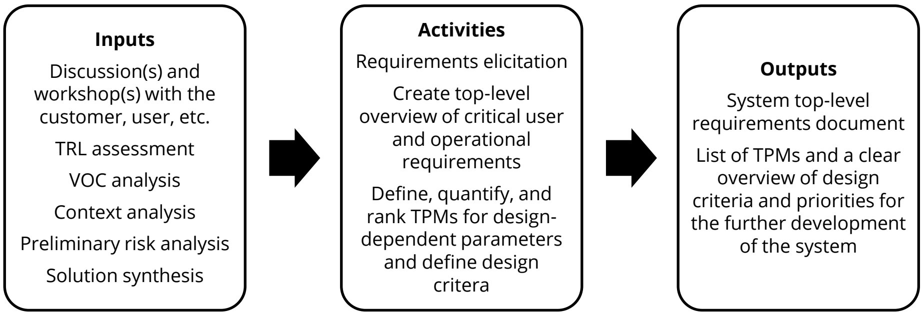


Figure 22. Requirements and TPM Analysis IPO diagram

6.2.3. Perform functional analysis and allocation

The SE function shall perform a functional analysis of the system and create a Functional Baseline and Allocated Baseline.

A good Functional Baseline is essential as it concretises the system top-level objectives and it gives an overview of the system interfaces and requirements. This baseline serves as a common frame-of-reference for the many engineering activities down the line. It shall be reviewed internally and by the customer before requirements are allocated to the next lower (discipline) level.

Work done properly at this stage ensures the further design team is aligned properly and working towards fulfilling the right goals. Since this functional analysis is conducted at system level it is the responsibility of the SE function. While the SE function may not accomplish the functional analysis in total, it holds a strong leadership role notwithstanding.

Requirements are allocated to the next lower level in an Allocated Baseline. This is where the system functions are distributed over its constituents and divided among the HW/SW/ME disciplines. The systems engineer shall work together with the discipline-specific functions to ensure that the top-level system requirements are linked properly and are traceable to the lower-level discipline requirements.

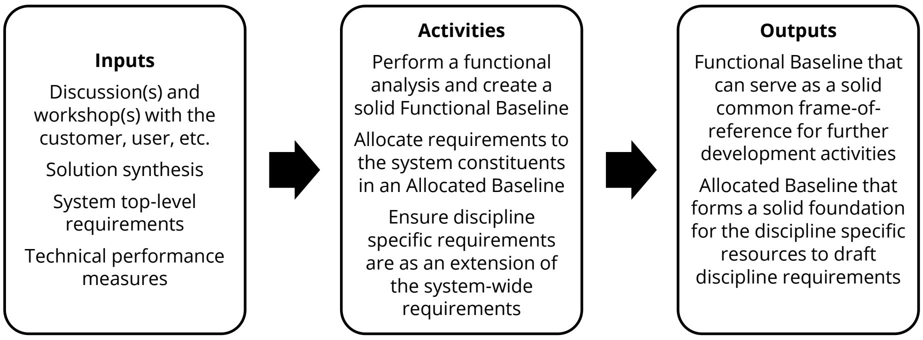


Figure 23. Functional Analysis and Allocation IPO diagram

6.2.4. Create a Concept Design Description

A document shall be created that summarizes the previous activities and serves as a common frame-of-reference for the subsequent engineering activities. It is also a key document to be approved before the release of the concept. It includes the following:

- Results of solution synthesis
    - Problem statement
    - VOC analysis
    - TRL assessment
    - Feasibility studies
    - Proposed solution
- Results of requirements analysis
    - Critical system requirements
    - Design considerations
    - TPMs and derived design criteria
    - Top-level system requirements specification
- Results of functional analysis
    - Functional Baseline (functional decomposition of the system)
    - Allocated Baseline (functions allocated to discipline)

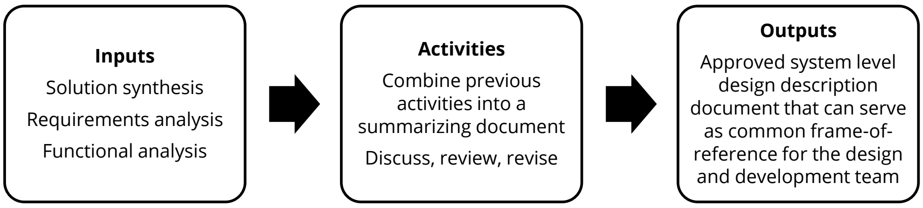


Figure 24. Concept Design Description creation IPO diagram

6.2.5. Create and maintain a System Architecture Overview

For subsequent development activities it is important that all engineers in the project share a single source-of-truth. For this, the SE function shall spearhead the creation and maintenance of a System Architecture Overview document. An appropriate approach for this may be the A3 Architecture Overview (A3AO) (see Section 7.2.7) [44] [45] [46], but other forms are possible.

Regardless of its form, several requirements hold true: It is important that this System Architecture Overview is easily accessible to all project staff. Furthermore, it shall be at a level of detail and complexity so that it is easily understandable for all (not only systems engineers), but still comprehensive. Lastly, special care should be taken to ensure that it remains up-to-date and consistent as the project evolves. If any of these requirements are not met, discipline engineers may operate and make design decisions based on inadequate or outdated system representations.

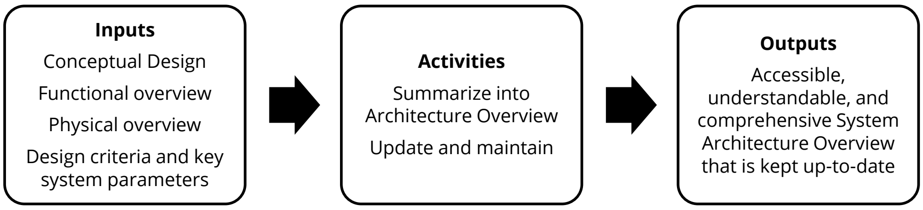


Figure 25. System Architecture Overview maintenance IPO diagram

6.2.6. Support Verification and Validation Plan creation

The SE function shall assume a leadership role and provide support for defining acceptance criteria and appropriate tests for the various verification activities over the system life cycle. It is important that system testing properly reflects the initially defined critical characteristics and measures of effectiveness, and that there is connectivity between the prioritised TPMs, requirements, and the various (discipline) design aspects. The SE function plays an especially important role in system reviews, as not all acceptance criteria can be covered by formal testing.

For all levels of system verification (testing, review), the SE function must enable the creation of a viable test and evaluation approach and ensure that it is properly connected to the various system specification documents. Finally, while validation is customer responsibility, the SE should ensure that the delivered product meets validation expectations.

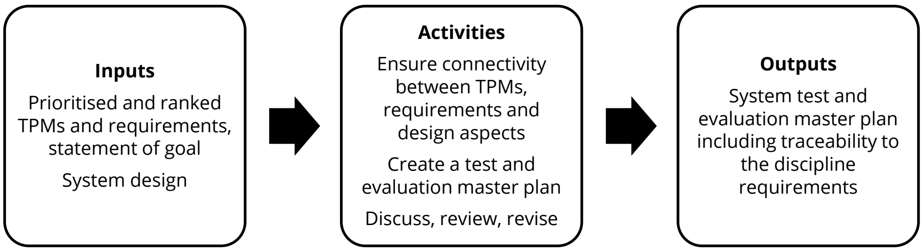


Figure 26. Design V&V creation IPO diagram

6.2.7. Support Synthesis, Analysis, Evaluation

Although this activity is continuous and performed by all design disciplines involved, the SE function must provide continuous oversight and guidance. Based on the complexity of the design project, the involvement of the SE function may vary and in any case involvement shall be flexible. The key goal for the systems engineer is to ensure that all day-to-day development efforts are in compliance with the system specification and work towards accomplishing the goals as prioritized. Design trade-offs and engineering decisions shall be documented clearly for future reference and design risks shall be monitored and addressed (see also 6.2.8 and 6.2.11).

The form that this activity takes on varies greatly per project and is strongly dependent on technological risk, team size, and complexity and size of the development effort. In projects with high technological maturity and low risk, where the development efforts are relatively straight-forward, it may be sufficient for the systems engineer to solely be part of the design review panel. However, for larger, more complex projects closer leadership may be necessary. In these cases the SE may be a structural part of the weekly (or more frequent) internal progress meetings.

In any case, the systems engineer should aim to keep the design team focussed on solving the right problem, within budget and planning, and should prevent scope-creep. The specifics of this activity and level of involvement of the systems engineer shall be specified at the start of the project in the DDP.

Figure 27. Synthesis, Analysis, Evaluation support IPO diagram

6.2.8. Support Risk Management activities

Risk management is a continuous and forward-looking process. While it can be regarded mainly as a project management task, it is also part of SE. The goal of the SE function for this activity is to support project management in anticipating and averting (technical) risks that may adversely impact the project.

Broadly, this activity involves 1) defining a risk management strategy, 2) identifying and analysing risks, 3) handling selected risks, and 4) monitoring the progress in reducing risks to an acceptable level [41] [47] [48]. For this, at the start of the project, a risk management strategy shall be established in the form of a Risk Management Plan or similar.

Risks shall be identified continuously over the project at system level as well as subservient levels. As the project progresses and more detailed design occurs, risk identification shall be repeated. The SE function shall be responsible in risk elicitation and analysis at system level, and coordinate this for subordinate levels.

There is no one way to perform risk analysis, but a popular method is to create a Risk Matrix in which risks are mapped based on likelihood and occurrence. The results of this risk ranking shall be discussed with and accorded by the customer and relevant stakeholders. It serves as input for the subsequent engineering activities, as well as for System Testing and Evaluation.

Risk mitigation shall be performed for all identified risks, systematically evaluating the probability of occurrence (likelihood) and consequence of occurrence (impact). Risks are mitigated by imposing additional requirements to lower likelihood or impact, or by changing the design. Ideally the risk is 'designed out', e.g. via the addition of an early-warning system.

As the project progresses, the risk elicitation, analysis, and handling activities shall be repeated at the detail level of the project. For instance, as part of the total risk management activity, in the Conceptual Phase of the project a Concept FMEA may be conducted, and later in the Design Phase the team performs one or more further detailed Design FMEA(s).

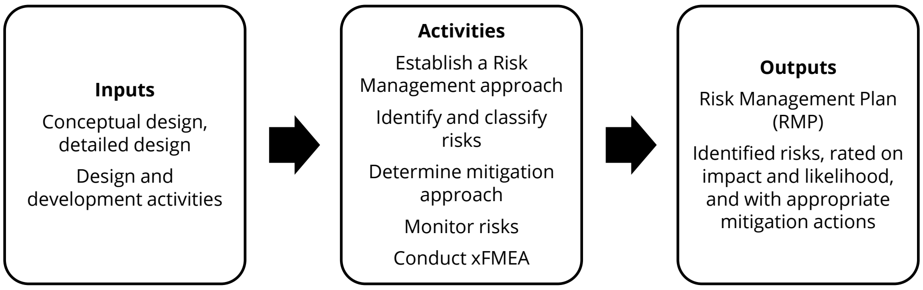


Figure 28. Risk Management support IPO diagram

6.2.9. Plan, coordinate, and conduct formal Design Review meetings

Formal design reviews are necessary to ensure that the design of the system satisfies the requirements set. At suitable stages over the system life cycle, formal design reviews are conducted, to check if the activities up to that point are in line with the specifications. These reviews form the stage-gates for progressing to the next design phase.

Design reviews typically include the following topics:

- Evaluation of the results of system design and development activities according to the defined requirements.
- Identification of problems and necessary actions.
- Closure of action items, anomalies, deviations, etc. identified as part of a previous design review.
- Updated risk assessment.
- Authorization for progress to the next stage.

Figure 29. Coordination of Design Reviews IPO diagram

6.2.10. Lead integration and monitor testing activities

This activity comprises three important parts of the system design and development process: system integration, testing (verification), and validation. System integration is the process in which the individual units/components are combined to form the whole system. After system completion, verification testing is performed to verify quantitatively that the system meets its requirements. Finally, validation testing ensures that the system meets its goals.

Verification of the system occurs at multiple levels of system completeness (Figure 30): the components individually, constituents of the system combined and integrated into subsystems, and finally the complete system.

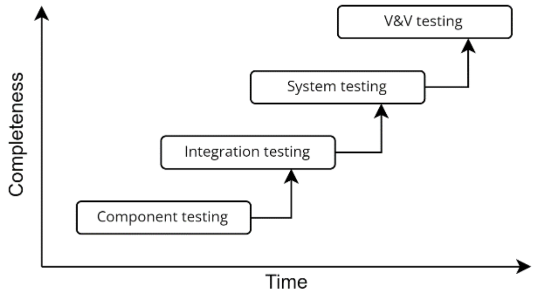


Figure 30. System testing levels

A proper system integration approach is necessary to ensure that the system constituents can effectively work together. Integration testing therefor occurs to expose defects in the interfaces and integrations between the subsystems. The SE function shall take a leadership role in this, and a system interface specification (or similar; may be part of system requirements specification) shall be leading.

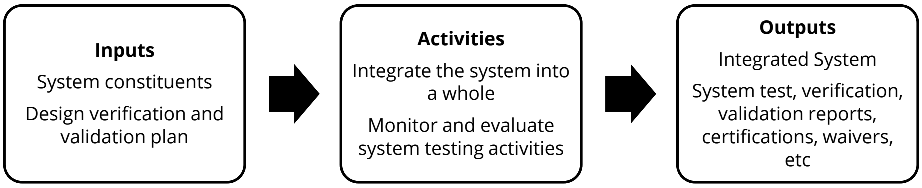


Figure 31. Testing support IPO diagram

6.2.11. Ensure proper documentation

After release of the concept/system design, a multitude of design and engineering decisions and trade-offs are made at discipline level over the course of the remaining design process. While the SE function is not responsible directly for the decisions made, the systems engineer must take on a leadership role to ensure that proper documentation is maintained. It is important to note that documenting only the *results* of trade-offs is not sufficient; the *rationale/reasoning* is equally as important and should be noted as well.

Especially for complex, large long-running acquisitions, proper design documentation is paramount in knowledge retention as the design team may change over time. Furthermore, documenting the rationale behind the decisions made inherently forces the design team to follow an academic approach, which benefits the quality of the end-product.

The specific way and level of detail to which this documentation shall be held shall be specified at the start of the project in the DDP, see also Section 6.4. For some projects, this document may also need to be included in the DHF, and/or transferred to the customer after project closure, and therefore must be subject to configuration management and reviews like any other official project documentation.

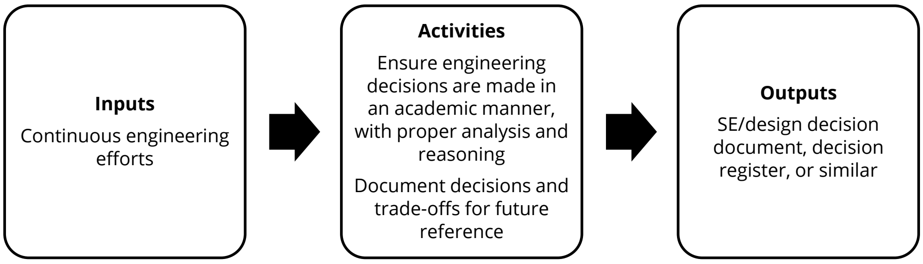


Figure 32. Documentation Control IPO diagram

6.3. Updated Benchmark life cycle model

As discussed in Chapter 4, Benchmark employs its own design control methodology process (the BE-11003 7-step life cycle model [36]). It is possible to map the above defined Systems Engineering program tasks on Benchmark's existing process. For this, both BE-11003 and the SE Program Tasks are summarized in an overview. By creating this integration, the role and responsibility of the systems engineer in a Benchmark project becomes clearer.

Figure 33 on the next page visualizes the Benchmark 7-step process, listing in three rows the general phase activities taken from BE-11003, milestones, and the main phase deliverables. A fourth row is appended that includes the SE program tasks as defined above. Benchmark-defined phase-deliverables are linked (dotted line) to SE program tasks when the SE function directly contributes to it. The start and end of the activities is visualized, although this may vary depending on the project.

The SE activities mainly fall into the earlier stages of the project. This is where the most unknowns are still present, and technical leadership is required by the sub-disciplines. As the project progresses into the later stages, the SE tasks transition into supporting the disciplines and maintaining up-to-date and cohesive documentation.

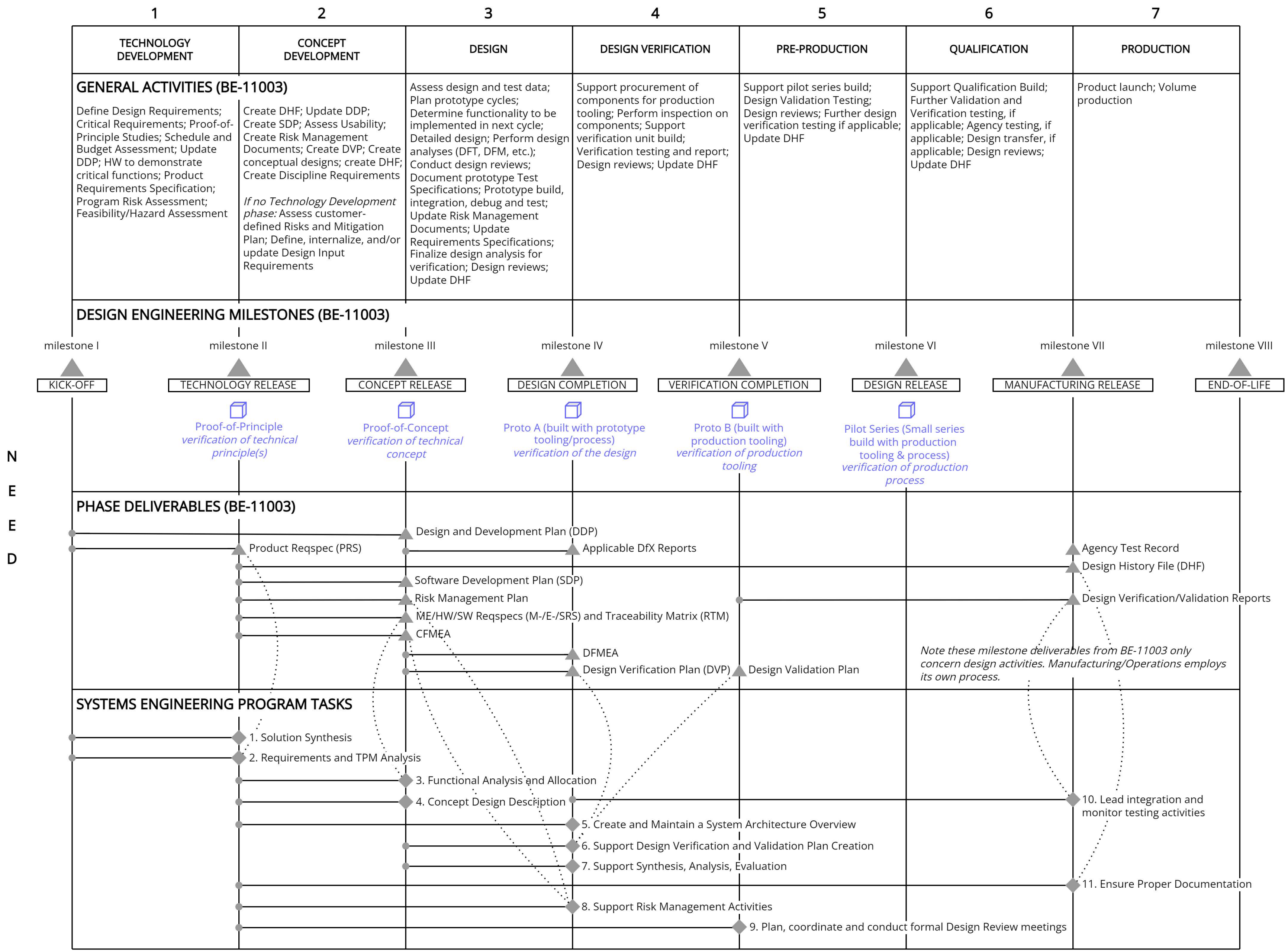


Figure 33. Benchmark Integrated SE approach showing SE program tasks integrated into BE-11003

Legend: circle = start point of activity/deliverable, triangle = end point of deliverable, diamond = end point of activity

6.4. Defining implementation level-of-detail per project

While the SE responsible program tasks have now been defined, it is challenging to formulate a single universally suitable level of detail for the implementation of these tasks. It is undesirable to impose a single set of rigid results, outputs, and deliverables onto all types of projects. For instance, a small low-risk acquisition has different requirements for documentation than a large high-risk medical 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.

It does not make sense to create a quantifiable set of guidelines or draft a comprehensive decision support system for this, because of the diverse nature of Benchmark projects. Not only is there variation in size, scope, and complexity, but different clients often impose their own requirements and way-of-working. Furthermore, depending on what standards the process must adhere to (ISO 9001, 21 CFR Part 820, ISO 13485, AS 9100), different requirements are also set. This leaves the level-of-detail best defined by the PM.

Nevertheless, next to the above mentioned, it is possible to draft some complexity indicators for the PM and principal design engineers to use in their decision[8]. These are intended as food-for-thought and are listed below:

1. **Technological Readiness Level of the technology applied in the project:** When there is a low TRL, and thus high technological risk, more SE involvement is desirable to increase the chance of project success.
2. **Size and complexity of the development effort:** In large, complex development projects, more SE involvement is desirable to increase the chance of project success. Projects with lower complexity require less involvement.
3. **Size of the development team:** In large teams it is difficult to ensure effective communication and cohesiveness of information. In the case of large teams, more SE involvement is desirable to ensure a single source-of-truth.
4. **Project budget:** Research suggests that complex engineering projects have the highest chance of success when SE expenditure is around 15% of total program cost [49]. It may be beneficial, depending on the project, to target 15% SE expenditure.
5. **Project market classification:** The risks inherent to the market classification (e.g. medical, aviation, industrial) of the deployed product inherently influences the design process and therefore the degree of SE involvement.

6.5. Organisational structuring within Benchmark

There are several ways of structuring a company and the disciplines within it. For instance, a function-based structure departmentalizes the organization based on common job functions. Another is a product-based structure, where each division in the organisation is dedicated to one particular product (line). Lastly, a matrix-based structure combines the two: employees report to their discipline leader, but also to the leader(s) of their project(s).

[8] While it is enticing to use these pointers as a starting-point for a formal decision-support-system (e.g. via a weighted Pugh Matrix with Likert-scale scores for these five points) it was deliberately chosen not to. While quantifying these points may be possible, it is the author's believe that effectively weighing these with regards to the outcome decision, is impossible or superficial at best. Furthermore, determining cut-off values for inclusion/exclusion of deliverables is impossible, as qualitative factors play as much a role as quantitative ones. The Benchmark PM has vast experience, knowledge of previous projects, and a direct connection to the customer and the project teams, and is therefore better suited to make the decision based on their own experience and assessment of the project.

The Design Engineering department of Benchmark can be classified as a matrix structure: employees have functional reporting duties to their discipline manager (Competence Leader), and product-based duties towards the PM of the project(s) they are involved in. Figure 34 visualizes the competence groups (including the new SE group) as pillars within Design Engineering. The coloured square, triangle, and circle icons in the figure represent different projects each with different tenure (full-time equivalents [FTE's]), showing an individual may be spread over multiple projects at the same time.

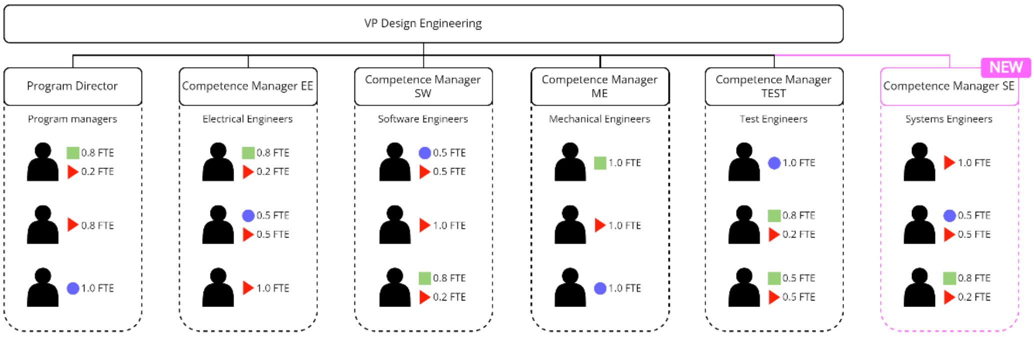


Figure 34. Benchmark Design Engineering as a matrix organisation

The columns in the figure represent the engineering disciplines (EE, SW, ME). The disciplines are managed by individual competence managers, who have responsibilities ranging from resource management to knowledge retention and training, recruitment, quality-control, etc.

Up to this point, in many cases the typical SE activities within any project were done by one of the traditional disciplines in an implicit SE role. In some projects, an explicit extra role was created for the systems engineer or systems architect, however it was always filled by one of the traditional discipline engineers: e.g. an electronics systems engineer or software systems engineer. A separate SE *competence group* within the company does not exist.

To fill this void, it is proposed to establish a new competence group for SE. This allows for flexibility of project assignment of the systems engineers, as their involvement in a project should change as it progresses. Similar to the existing competence groups, a competence leader should be appointed to ensure continuity in the department and should become process owner of the SE process.

The creation of the new competence group also represents a shift of some individuals within the existing structure. Currently, both the SW and EE group include discipline-specific systems engineers (classified as either SW/EE system engineer or system architect). These roles shall be moved to the new competence group, as this is necessary to solidify the transdisciplinary nature of the role.

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 (BE-11003), 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.

This diagram was designed based on the existing EE, ME, SW Processes and the Engineering Design Control Methodology. It was presented in the visionary focus group meeting consisting of system architects, competence leadership, and lead engineers. It also underwent both an informal walkthrough and one formal internal review.

The goal of this diagram is to show explicitly what activities and deliverables in a project are the responsibility of the SE function. The diagram also explicitly states when the outputs of an activity are handed over to the other disciplines, to be used as an input for the respective discipline actions, or when it takes input from the disciplines.

It is important to understand that the SE process must not be seen as strictly separate from the other discipline processes. While the process diagram explicitly lists 'hand-overs' of activity outputs and inputs, many activities should occur largely concurrently, with frequent communication between the disciplines.

Furthermore, the form of the process suggests that the activities occur in sequential, linear order. In reality, this is not always the case and loopbacks do occur. For instance, the *Design Review* activity occurs multiple times and at various stages in the project lifecycle. The same goes for the *Design Verification and Validation* activity. The choice to present the SE process in the form of this linear process flow is solely made so that it matches the already existing engineering processes in place (see Section 4.2). It may be worth looking at reworking the existing discipline processes as further steps are taken to implement this new SE process.

The following sections describe the process diagram in detail, separated into activities, and split over Figure 35-43. Thus Figure 35-43 combine into the full process. The full SE process in Benchmark's format (including the diagram) can also be found in Appendix E.

7.1.1. Kick-Off

During this activity the systems engineer assists the PM in the kick-off of the project. The team is formed and initial budget and planning are assessed. Furthermore, in this activity, the level-of-detail of the SE approach is determined (see Section 6.4) and recorded as part of the DDP.

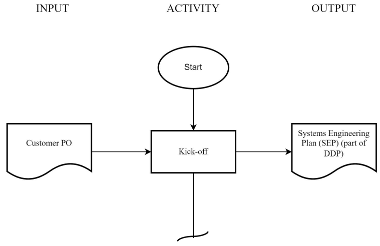


Figure 35. Kick-off

7.1.2. Solution Synthesis

In this phase the systems engineer explores the problem space and creates an initial solution direction. As part of this activity, the VOC and system context are analysed. The SE determines a solution direction and conducts feasibility studies and proofs of concept to investigate the feasibility of the solution and the TRL. The other disciplines may be involved in these activities as the systems engineer sees fit. The proposed solution is used by the disciplines in their respective processes.

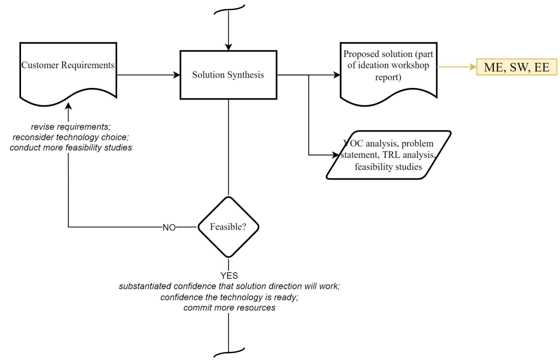


Figure 36. Solution synthesis

7.1.3. Requirements and Functional Analysis

This activity concerns the drafting of a system level requirements specification, known internally as the Product Requirements Specification (PRS). The systems engineer shall also collect any external requirements set by regulations or standards.

Next, functional modelling and analysis is performed in order to allocate functions to subsystems and assign responsibility to the subdisciplines.

Also a part of this activity is the creation of TPMs (measures of effectiveness), which drive design considerations down the line. System interfaces are described either as part of the PRS or a separate System Interface Specification (SIS) or Interface Requirements Specification (IRS). The draft PRS and allocated baseline are used by the subdisciplines in creating discipline level requirements (and concepts) in the next phase.

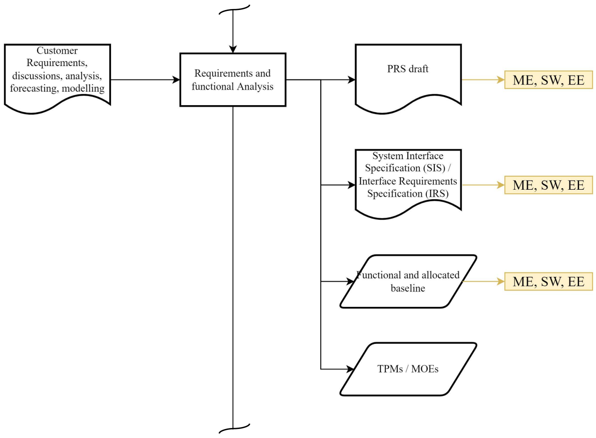


Figure 37. Requirements and Functional Analysis

7.1.4. Concept Development

In this activity the overall system concept is developed and approved in the form of a Product Concept Description (PCD).

As part of this, initial risk management is performed to inform the definition of the system concept. Additionally, as the concept is approved, a system architecture overview is created and from here on maintained to serve as single source-of-truth for the development team. Input to this activity are all previous analyses, test data, and design experience.

This activity works together closely with the disciplines since the discipline-level Concept Descriptions (E/S/MCDs) must match the PCD.

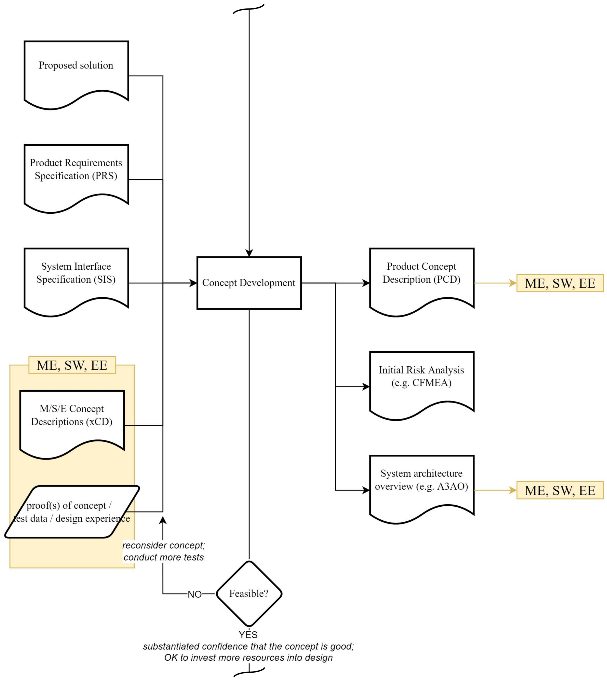


Figure 38. Concept Development

7.1.5. Requirements Specification

This activity expands on the initial requirements and functional analysis, by combining it with the final approved concept, finalizing the PRS.

Together with the final PRS, the system level Product Verification Plan (Test Plan) is created, along with requirements traceability documentation. Further risk management activities may also be performed.

Again this phase works closely together with the disciplines since the M/S/E Requirements Specifications, Traceability Matrixes, and Test/Verification Plans must match the System Level documents and 'talk to' each other.

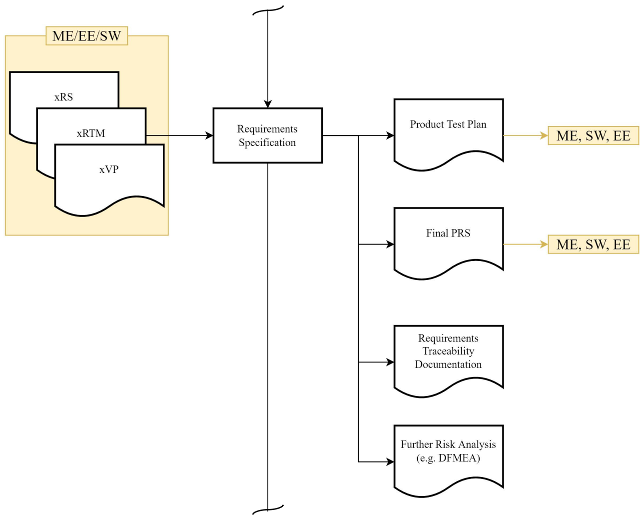


Figure 39. Requirements Specification

7.1.6. Detailed Design

This activity is cyclic, as the system is detailed more in-depth and as part of this activity the system implementation will progress through several baselines. For this a proper Configuration Plan must be established. Any changes made to the system must pass through the CCB, of which the systems engineer is a part.

The main tasks of the systems engineer in the Detailed Design phase are documenting design choices and assisting the disciplines with trade-off studies, taking into account input from the PCD, PRS, and constraints from the disciplines. All is recorded in the PDD, which is created in this phase.

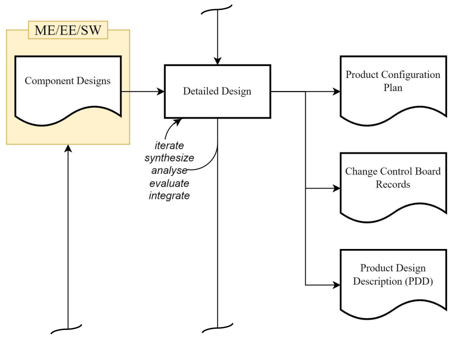


Figure 40. Detailed design

7.1.7. Design Review

Design reviews are performed at the end of each project phase, or intermittently at appropriate project milestones. The purpose of the design review activity is to provide a systematic assessment of design results and to provide feedback to designers on existing or emerging problems. For this, the systems engineer shall always be a part of the review team.

The outputs of this activity are Review Records (responsibility of the TPM), further risk analysis documents (e.g. DFMEA), but also a SE Acceptance Report, in which technical issues that arise during the project are addressed and resolved.

The SE Acceptance Report describes per issue its cause and appropriate actions to prevent or correct it. This may be aimed at prevention, correction, or improvement. It is maintained over the multiple Design Review activities in the project life cycle.

The maintenance of the SE Acceptance Report is part of a broader SE Acceptance and Control process which involves reviewing and prioritizing potential action plans together with the TPM and review panel and analysing interrelated issues to ensure a consistent and cost-effective resolution to problems that arise during the design.

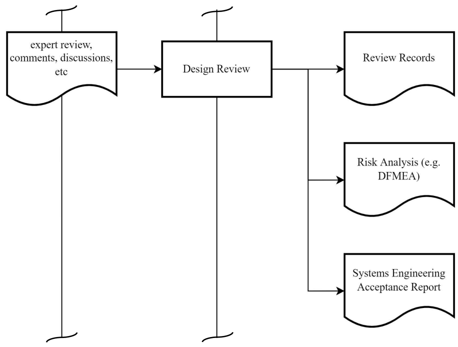


Figure 41. Design Review

7.1.8. Design Verification and Validation

This activity comprises both the verification and validation of the designed system. The SE is responsible that the system level design matches the specified requirements. They also assist the disciplines in their respective discipline-level verification activities.

It is important to understand that System Verification and Validation is a transverse activity and should be applied to every stage of the system's life cycle, including during the development cycle. This means that the verification process should be conducted in parallel with the processes of defining and realizing the system. The verification process should be applied to all activities and products that are produced as part of the development process, regardless of the stage of the life cycle in which they occur. This is important because it helps to ensure that the system meets all necessary requirements and functions as intended, both during development and throughout its lifetime.

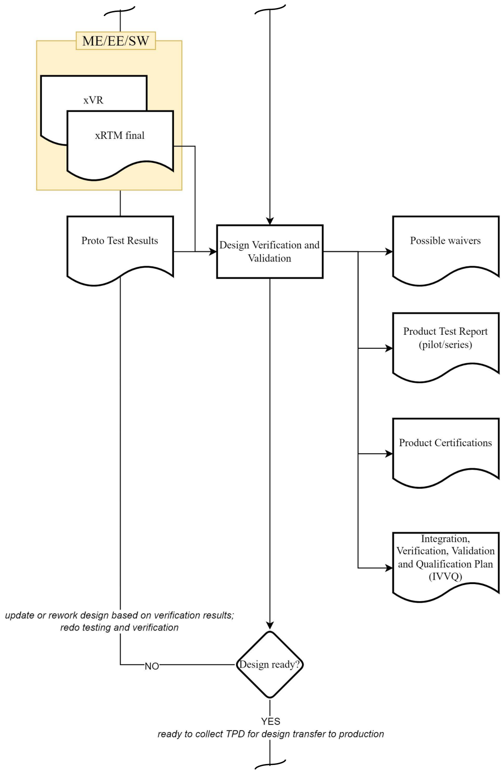


Figure 42. Design Verification and Validation

7.1.9. TPD Check

This activity occurs in the final stage of the project before the Design Release and transfer to New Product Introduction (NPI) Engineering. While it is the responsibility of the TPM to collect and properly archive the DHF, the systems engineer has an important role in ensuring technical comprehensiveness of the TPD that will be archived in the DHF.

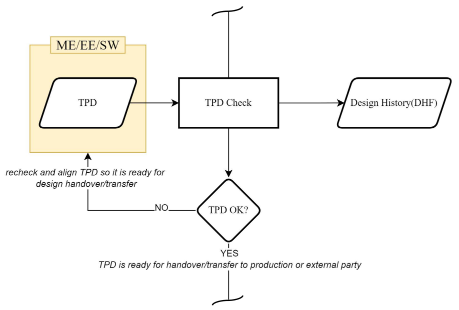


Figure 43. TPD Check

7.1.10. Design Release and start of Life Cycle Management

In this activity the design is released for production, and the project is concluded for Design Engineering. The systems engineer and TPM have a role here in performing a retrospective to assess the technical considerations made and the applied processes. Future improvements should be noted and made readily available as design input for new projects.

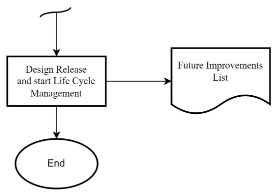


Figure 44. Design Release and start Life Cycle Management

This activity also includes the Life Cycle Management tasks of the systems engineer, which include supporting NPI Engineering and ensuring proper support over the system life cycle, including end-of- life. The specific form and level of involvement of the systems engineer in Life Cycle Management tasks is dependent on the nature of the project.

7.2. Project deliverables

7.2.1. Systems Engineering Plan (new)

The new definition of the SE Plan (SEP) is necessary to clearly assign responsibility of technical management to the systems engineer. The definition of many parts of the DDP that already exist (see Section 4.4.1) will come to fall under SE responsibility, or at least the systems engineer shall have a strong supervising role towards the PM.

Three options exist for the introduction of the SEP. Following thorough internal discussion about how the SEP shall be formulated it is deemed helpful to explicitly state the three options. They are the following:

- Keeping the current form of the DDP, shifting responsibilities of some parts to the SE. A benefit of this approach is that the form of the DDP will not change, allowing the use of current templates. However, this will convolute the divide in responsibilities of what parts fall under the PM and what parts belong to the systems engineer.
- Separating the DDP into two documents: a Program Management Plan (PMP) and the SEP. This allows for a clear divide in responsibilities, however requires the introduction of two new documents which may limit adoption.
- The DDP is kept as a singular document, but is split into two parts: PMP and SEP. This combines the best of the above two options and therefore preferred. This is also similar to the new Software Development Plan Procedure.

The SEP addresses the strategy and approach for all technical aspects of the project, including the system architecture and development, risk management, quality management, configuration management, information management, verification and testing, integration, validation, production, and deployment. SE planning should account for the full scope of technical activities and ensure that they are integrated in order to achieve a comprehensive and integrated plan for the project.

It is new document, reviewed by a team including the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

7.2.2. Proposed Solution Document (new)

As part of the *Solution Synthesis* activity, a solution direction is determined. This direction (or directions) shall be documented in a new Proposed Solution Document.

Within Benchmark the proposed solution in a project is already sort of documented as part of the reported output of the Creative Workshop that is held at the start of all development projects (see Section 4.1.1). However, formalizing this step in a new document is desirable.

The new Proposed Solution Document shall comprise general subsystem lay-out and the initial direction for the technological solution to the identified problem. Depending on the complexity of the project it may also include a VOC/critical-to-quality analysis, TRL analysis, and initial feasibility studies to support the TRL verdict. The level-of-detail for this document is low: it should be enough to allow the creation of an initial schedule and budget, but it may change in the concept development phase of the project.

It is a new document, reviewed by the PM, systems engineer, relevant discipline lead engineers, and the customer product manager and other representatives like marketing, and sales.

7.2.3. Product Requirements Specification

The PRS is an existing document that lists all product (system) level requirements and serves as the result of the translation of customer requirements. It is used as input for the applicable disciplinary (EE, SW, and ME) Requirements Specifications, and it is intended to explicitly describe what the product should do.

The requirements described in the PRS serve as an agreement with the customer and are divided into two groups: Functional and Non-Functional requirements. The Functional requirements are expressed as use cases and scenarios involving certain actors in the different subsystems, while the Non-Functional requirements relate to things like performance, environment, standards, and usability requirements.

The PRS shall specify what the system shall do, but not how it should be done. For example, it may specify the system's performance in terms of the number of operations per second, but it will not specify which microprocessor should be used to achieve this performance. The system will be verified during and at the end of the design process based on the PRS, for example by testing whether the actual number of operations per second matches the specification in the PRS.

No changes are made to the PRS procedure by this thesis and the PRS is not a new document. The PRS is reviewed by a team including the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

7.2.4. System Interface Specification or Interface Requirements Specification (new)

For some complex projects it can be beneficial to create a new separate requirements document for the requirements for interfaces between subsystems, and for the interfaces between the system and its environment. The purpose of the SIS or IRS is to provide a definition of the product interfaces (external and internal). This includes user, electrical, mechanical, software, optical, thermal, and other interface parameters and connections.

This is a new document (SIS or IRS)or could be part of the PRS (determined the PM and systems engineer determine at the start of the project). It is reviewed by the PM, systems engineer, EE and SW Architects and lead engineers, and the customer product manager.

7.2.5. Product Concept Description (new)

The PCD is the output of the Concept Phase. It redefines the initial Solution Direction into a more refined and final concept to enter the Design Phase with. The PCD is approved by the design team and the customer and finalized/locked when the project exits the Concept Phase. The PCD is input to the PDD, which is created and maintained over the Design Phase.

The PCD is a new deliverable and it is reviewed by the PM, systems engineer, relevant discipline lead engineers, and the customer.

7.2.6. Risk Analysis Documentation

In the design process risk management activities are performed at various stages and levels (e.g. CFMEA, DFMEA, FMEA). The systems engineer is responsible for the creation and maintenance of risk management documentation over the system life cycle. There should be compliance to BE-11008 [50]. From the risk management documentation it should be clear 1) what risks have been identified, 2) their ranking, including approval by the customer, and 3) their mitigation in the form of what new requirements and design elements.

The CFMEA, DFMEA, FMEA, and other risk management documents are existing deliverables. The various risk analysis documents are reviewed by a team consisting of the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

7.2.7. System Architecture Overview (new)

As stated previously, in complex development efforts it is difficult to keep the whole team and everybody involved up-to-date with the current state of system design. This is not because design considerations are poorly documented, but because the amount of information in the documentation is often very extensive (as it should be), but therefore long, hard to read, and sometimes outdated (or WIP) [51].

The aim of introducing a new Architecture Overview document is to present an overview of the system architecture that can be used as a starting point for learning and a way to communicate the system architecture with stakeholders. It should not replace existing documentation, although in some cases it may, but instead present a focused overview of the current standing of the system design.

One such way is via Daniel Borches' A3AO, which aims to accomplish the above on one (double-sided) A3-size document [51] [52]. The limited size of the document forces the systems engineer to only include key documentation. It includes on one side of the paper a Functional and Physical breakdown, including a list of the most important Design Decisions. The other side summarizes the project via text in a condensed way: it includes the System Partition, Functional Flow, Physical Decomposition, System Concerns, Key Parameters, Design Guidelines, and a summary of the Development Roadmap. Figure 45 presents an example, more can be found on the site [53].

Note that this is not the definitive way in which Architecture Overviews should be created, but it may serve as a good starting point. The goal should be to create an easy-to-understand document that can be checked quickly and serves as a good basis for understanding the system design.

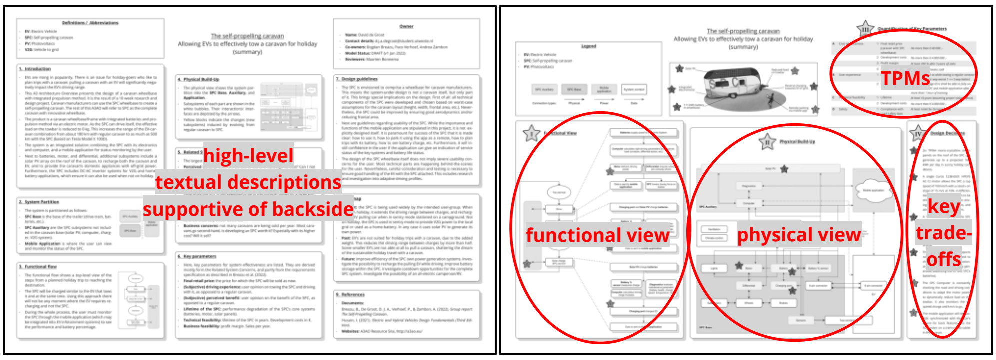


Figure 45. A3AO example front and back (reproduced from [54]) with legend

7.2.8. Product Design Description

The PDD is an evolved version of the PCD. It defines the system at a higher level of detail. Included in this document should be the System Definition with Operational Requirements (Need, Mission, Use Profile, Life Cycle), Maintenance Concept, Functional Analysis and Allocation, Interface Specification, Performance Measures, Physical and Usability (Human Factors) Characteristics, and Sustainability and End-of-Life Characteristics.

The document shall also describe Design and Construction characteristics, including CAD/CAM Requirements, Materials, Processes and Parts (BOM), EMC, Safety, and Testing Considerations, and data on Economic Feasibility. Furthermore, included should be

Logistics information such as Maintenance Requirements, Supply-Chain, Personnel and Training, Facilities and Equipment, Quality Control, Packaging, Handling, Storage, and Transportation Requirements, and Customer Service Considerations.

No changes are made to the existing PDD procedure. The PDD is reviewed by the PM, systems engineer, EE and SW architects and lead engineers, and the customer product manager and other representatives like marketing, and sales.

7.2.9. Product Test Plan / Design Verification Plan

The PTP or Design Verification Plan (DVP) documents the product-level testing to be performed for verification of the system. The plan should describe how all requirements are to be verified, in what time frame, and under what external conditions, including specifics like test equipment and settings. It should integrate with the discipline-specific test plans.

The PTP/DVP is reviewed by the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

7.2.10. Requirements Traceability Documentation

This deliverable consists of a system-level Requirements Traceability Matrix document that links discipline requirements to system-level requirements and tests. It shows the hierarchy of requirements and allocation of tests. It ensures that all customer requirements and derivatives are implemented and tested.

The traceability documents are reviewed by the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

7.2.11. Product Configuration Plan (new)

The Product Configuration Plan document is initiated during the concept phase of the product's development and updated throughout its life cycle. It records the desired configuration(s) of the product at the early concept phase and any existing configurations during subsequent phases. This plan serves as a reference framework for product configurations and variants, and it may be used as input for the development of electronics, mechanics, and software architectures, as well as the Product Test Plan and factory test requirements.

In addition, the Product Configuration Plan helps the project manager to maintain the development scope and ensure that major product variants are identified and included in the proposed solution at the proposal phase.

The Product Configuration Plan is reviewed by a team of professionals including systems engineer, EE and SW Architects and lead engineers, NPI/setup and test engineering, and the customer product manager and other representatives like marketing, and sales.

7.2.12. Change Control Board Records

The CCB is a group that is initiated after the project enters Change Control. Change Control is a process that is initiated after major design verifications are started to prevent the need for these verifications to be repeated, which can cause budget constraints. The CCB may continue to operate as long as required by the program management.

The CCB Records contain a summary of the items that have been discussed, the decisions that have been made, and any open action items. The meeting minutes from the CCB are reviewed by all members of the CCB.

7.2.13. Systems Engineering Acceptance Report (new)

The SE Acceptance Report (SEAR) is a document that provides an overview of the compliance of the design (before it is realized). It identifies the compliance of the design with customer requirements and may be accompanied by an updated customer requirements matrix. The focus of the SEAR is the assessment of whether or not the design meets the required compliance.

If the compliance itself cannot be provided yet (e.g. due to the design's EMC behaviour), then a risk assessment of those requirements is included in the SEAR. The document may also include a description of the design's reusability.

The SEAR is reviewed by a team including the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

7.2.14. Product Test Report

The Product Test Report is a document that gathers all relevant information after the Realization Phase (e.g. Proto A/B) and the Verification and Qualification phases (e.g. Pilot/Series). It contains all test results and summarizes the configuration baseline, open issues and problem report identifications, agreements on incompliances, and the coverage of customer requirements.

It is reviewed by all involved discipline lead engineers, members of the CCB, and the customer.

7.2.15. Waivers

Waivers are written to obtain formal approval from the customer for deviations in the design process. Waivers are used when a project plans to deviate from agreements or requirements at any stage, including the requirements verification phase and the pre-production/pilot series. A waiver is a request for formal approval of a deviation that has already occurred, while a deviation request is a request for approval of a deviation that is planned for the future. They are usually reviewed by the quality manager or project quality engineer, PM, and customer.

7.2.16. Product Certifications

Certifications are formal proof that the product complies with the required regulations or stipulations present within the contract with the customer or imposed by an independent, national, or international organization. This deliverable summarizes declarations of performances such as external Test Reports and CE-marking descriptions and demonstrations of the fulfilment of the various quality assurance and performance tests. It is reviewed by the quality manager or project quality engineer, PM, and customer.

7.2.17. Integration, Verification, Validation, and Qualification Plan (new)

The Integration, Verification, validation, and Qualification plan (IVVQ) defines the final integration and verification strategies for the product. It is based on input from the SEP and DDP, as well as the conceptual design and requirements specification documents.

- The integration part of the plan describes how the different system components (coming from different disciplines) will be put together and tested.
- The verification part of the plan describes how the design will be verified, checking compliance with all requirements. The plan should describe how all requirements are to be verified, in what time frame, and under what external conditions. This is an evolved and final version of the PTP/DVP.
- The validation part of the plan describes how the system performance will be validated to ensure that it meets the customer's expectations and requirements.
- The qualification part of the plan outlines the qualification tests that need to be done on the design to meet regulatory requirements, and it also allocates resources, time, and locations (which may be external) for these tests.

The IVVQ is reviewed by a team including the quality manager or project quality engineer, PM, systems engineering, relevant discipline lead engineers, and customer representatives.

7.2.18. Future Improvements List (new)

As the product enters its life-cycle management phase after production, the systems engineer shall stay involved to monitor and index possibilities for future improvements. This is because after release, it is still possible for the product to be updated and improved. These improvements could be based on internal retrospective sessions, feedback from the customer or end users, changes in technology or market trends, or new features that were not included in the initial release.

By maintaining this list of potential improvements, the systems engineer can ensure that the product stays current and relevant over time. Additionally, this list can be used to prioritize and schedule future updates to the product, and to track the progress of these updates. It can help the systems engineer to identify any potential issues or challenges that may arise during the development of these updates, and to plan for how to address them. Finally, this list may also be useful during the sales trajectory for acquiring new projects.

7.3. 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 0.

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.

Table 13 on the next page shows how the activities and deliverables trace back to the SE program tasks defined in Section 6.2.

Table 13. Traceability of SE process activities and deliverables to related SE program tasks

Activity	Deliverable	Related SE Program Tasks[i]
Kick-off	▪ SEP	▪ 1, 8, 11
Solution Synthesis	▪ Proposed Solution Document ▪ VOC analysis ▪ Problem statement ▪ TRL analysis ▪ Feasibility studies ▪ Proofs of concept	▪ 1, 2, 8, 11
Requirements and Functional Analysis	▪ PRS ▪ SIS/IRS ▪ Functional baseline ▪ Allocated baseline ▪ TPMs	▪ 1, 2, 3, 11
Concept Development	▪ PCD ▪ Risk analysis CFMEA ▪ System Architecture Overview	▪ 1, 2, 3, 4, 5, 7, 8, 11
Requirements Specification	▪ Product Test Plan ▪ PRS ▪ Requirements traceability documentation ▪ Risk analysis DFMEA	▪ 2, 3, 6, 8, 11
Detailed Design	▪ Product Configuration Plan ▪ CCB Records ▪ PDD	▪ 5, 6, 7, 8, 9, 10, 11
Design Review	▪ Review Records ▪ Risk analysis DFMEA ▪ SEAR	▪ 8, 9, 11
Design Verification and Validation	▪ Waivers ▪ Product Test Report ▪ Product certifications ▪ IVVQ	▪ 6, 8, 9, 10
TPD Check	▪ DHF	▪ 11
Design Release and start of Life Cycle Management	▪ Future improvements list	▪ 9, 10, 11

[i] *For reference, the SE Program Tasks are: 1 = perform solution synthesis, 2 = perform requirements and TPM analysis, 3 = perform functional analysis and allocation, 4 = create Concept Design Description, 5 = Create and maintain System Architecture Overview, 6 = support Design 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 verification activities, 11 = ensure proper documentation. For more detail refer back to Section 6.2.*

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] [55]:

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.

A big part of Change Management is a stakeholder analysis. The goal of this is to identify all stakeholders affected by the change, including external stakeholders like customers and other partners. A proper stakeholder analysis allows the change leader to not only identify resistance, but also to form a team of the right people to pass on the vision and sense of urgency of the change.

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.

Of course, over the course of writing this thesis many stakeholders were already involved (e.g. through the SE Capability Assessment in Chapter 5). However, it is worthwhile to explicitly note the stakeholders. The list can be used to anticipate resistance and for targeted communication to ensure buy-in. The following stakeholders are identified, including their involvement so far:

- **Design Engineers:** Not yet involved.
- **Senior Design Engineers:** Shallow involvement for feedback on vision.

- **HW/SW Systems Engineers:** Fully involved in vision definition and progress meetings to discuss and verify direction.
- **HW/SW System Architects:** Fully involved in vision definition and progress meetings to discuss and verify direction.
- **Program Management:** Involved through SE Capability Assessment interviews to determine improvement areas for SE.
- **Discipline Leaders / Competence Management:** Indirectly involved through analysis of discipline processes. Involved in progress meetings to discuss and verify direction.
- **MT:** Involved only by proxy via the Discipline Leaders / Competence Management.
- **Test Engineering department:** Not yet involved.
- **Operations/NPI Engineering department:** Not yet involved.
- **Customers:** Not yet involved.

For the successful implementation of the new process, it is crucial to secure buy-in from the PM. The PM must fully support the new way of working and understand the reasoning behind it. This is because they need to sell the concept of a dedicated SE role to the customer, which may be challenging for existing clients. The PM must clearly communicate that, although the new process may initially increase project costs, it will ultimately save the client money by preventing mistakes, recalls, and improving overall quality.

Resistance to the change may also arise from the project teams themselves. The redefinition of responsibilities, particularly for the system engineer and architect roles, will require a learning curve for all engineers. This will likely result in significant resistance as many SE tasks will become explicit responsibilities of the systems engineer instead of being shared by the project team as a whole.

The organisation must expect a Kübler-Ross Curve (denial, anger, bargaining, depression, and acceptance [56]) at this level, since tasks will shift and the team must adapt to a complex new set of responsibilities. To address this, it is important to provide adequate guidance from Competence Leadership and Program Management to support struggling team members and to promote understanding of the rationale.

In short, 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

An important part of overcoming resistance is effective communication about the change, to ensure that all stakeholders understand the change and its rationale. Emotional reactions and resistance to change are inevitable, meaning proper communication about the change is paramount in ensuring its success. Making sure the purpose of the change is understood by all is most important.

For this, 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 Learning Management System (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.

Because this pilot comprises the SE program's V&V Phase, it should be assessed on its success. However, as demonstrated in Chapter 5, it is difficult to assess the SE effectiveness objectively and quantitatively. Instead, focus should be on evaluating the program's effectiveness in a qualitative way. It will be difficult to compare to current projects, or projects without the SE process in place. Nevertheless, the engineers that work on such development projects have sufficient insight to reflect on their own working methods.

With this in mind, evaluation shall be done both continuously and retrospectively. For instance, during the project the team may come together with a SE coach every month to discuss their way-of-working, challenges, and their interpretation of the new program. Key discussion points are the perceived efficiency of the project, changes from the old way-of-working, quality and effectiveness of communication and collaboration, and to what extent the newly defined responsibilities are clear.

Retrospectively the project shall be evaluated again. Here, some quantitative assessment may be possible, looking at project performance in terms of timeliness (schedule variance, delays, and their cause), financial performance (budget creep and its cause), and the budget allocated to SE. Qualitative assessment again should be done together with the team, but also with the customer. The focus should be on end result of the project and its quality and improvement points. Could they have been avoided? What choices led the team to this result?

All in all, executing one or more pilot projects is strongly advised before implementing any formal changes to Benchmark's way-of-working. The pilot project is crucial in refining the SE process and ensuring its effectiveness. It will also create a strong support group for change as the pilot team will have first-hand experience with the change. It is important to plan for success in the pilot project as chances for a second try are limited.

8.3. Implementation, training, and support

As has become clear from this chapter, to successfully implement the process it is important to explain the rationale behind it. While the pilot plays already a large role in this as the pilot team becomes an important change coalition, broad implementation of the program requires explicit training efforts.

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 46 on the next page presents an overview and the next sections give additional detail.

8.3.1. Systems Engineering Masterclass

SE is a complex profession. It not only requires a strong affinity with the technical disciplines, but also a good grasp of the engineering process and the ability for systems thinking. While future systems engineers should be selected based on their experience and knowledge about complex engineering projects, a SE Masterclass should be set-up to familiarize prospective systems engineers with Benchmark's SE process.

The Masterclass should be focused on what program tasks fall under the responsibility of the systems engineer, according to the process outlined in Chapters 6 and 7. Through this education, future systems engineers should become adept at the various tools and techniques that may be used, and in recognizing when to use them. This is especially important since the SE process does not prescribe the detailed execution of most tools, but only the rationale and expected learnings.

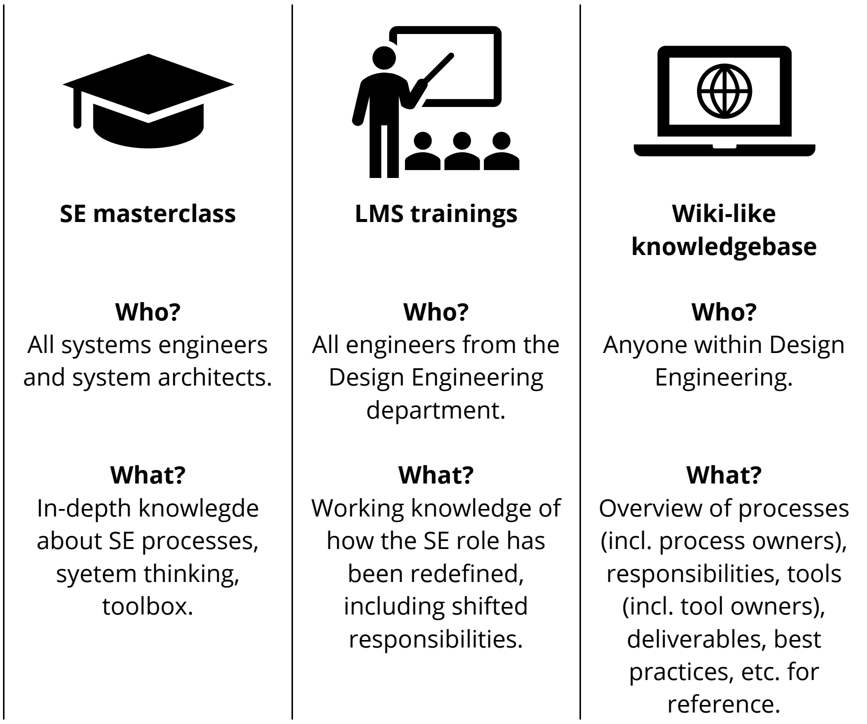


Figure 46. Types of SE training

8.3.2. Systems Engineering Knowledgebase

The various tools, techniques, and workflows should be documented in a SE Knowledgebase. The goal of this is to serve as a sort of internal Wiki for all relevant stakeholders. It should be less formal than approved engineering documentation and procedures, to allow the knowledgebase to be more focussed on the 'how' instead of the 'why.' It should also be a place where learnings and best-practices can be shared, so editing capabilities should be widespread.

8.3.3. LMS trainings

Benchmark has implemented a LMS to great success. The LMS is a web-based tool to instruct employees through text, video, and interactive elements (e.g. quizzing, testing). Any new employee is already prescribed several recurring trainings through this system, including trainings about the engineering processes. This makes it the perfect channel for an introduction to SE.

As stated before, it is important to educate all internal stakeholders on the changes caused by the new SE process. This ensures that all stakeholders are aware of the shifts in responsibility and the placement of the new SE role. A LMS training allows for more targeted and focused learning than via other channels like email and the weekly newsletter. However these channels may include references to the LMS to highlight the importance of the training.

8.4. Monitoring and continuous improvement

An important part of proper implementation and adoption of the new program also comes from aftercare: the new SE competence group should continually monitor, evaluate, and improve the program.

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.

To continuously refine the SE process, it is important to communicate the results of the monitoring and evaluation process to the stakeholders, including any successes, challenges, and areas for improvement. Additionally, organizing retrospectives after project completion can provide valuable insights into what worked well, what could have been done better, and what can be improved for future projects. This process can be similar to the interviews and retrospectives presented in this thesis.

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 0, 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, "BE-11003 Engineering Design Control Methodology, Rev. H," Benchmark, Global, 2022.

[37] Benchmark, "AN-11020 Electronics Engineering Process, Rev. 4.0," Benchmark, Almelo, 2021.

[38] Benchmark, "BE-11014 Software Engineering Process, Rev. E," Benchmark, Global, unknown.

[39] Benchmark, "AN-11015 Mechanical Engineering Process, Rev. 3.0," Benchmark, Almelo, 2015.

[40] 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.

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

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

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

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

[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] 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.

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

[49] 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].

[50] Benchmark, “BE-11008 Medical Risk Management Procedure, Rev. F,” Benchmark, Global, 2020.

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

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

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

[54] 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.

[55] 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.

[56] 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].

[57] 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.

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

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

[60] 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].

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

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

APPENDIX

APPENDIX A — EE PROCESS

Benchmark electronics

BEA Process Description
Electronics Engineering Process

QMS Document Number: AN-11020
Revision Level: 4.0

2.1 Process flowchart

Figure 2.1 shows the electronics engineering process flowchart. In the following chapters the activities with their inputs, outputs and responsibilities are described in detail.

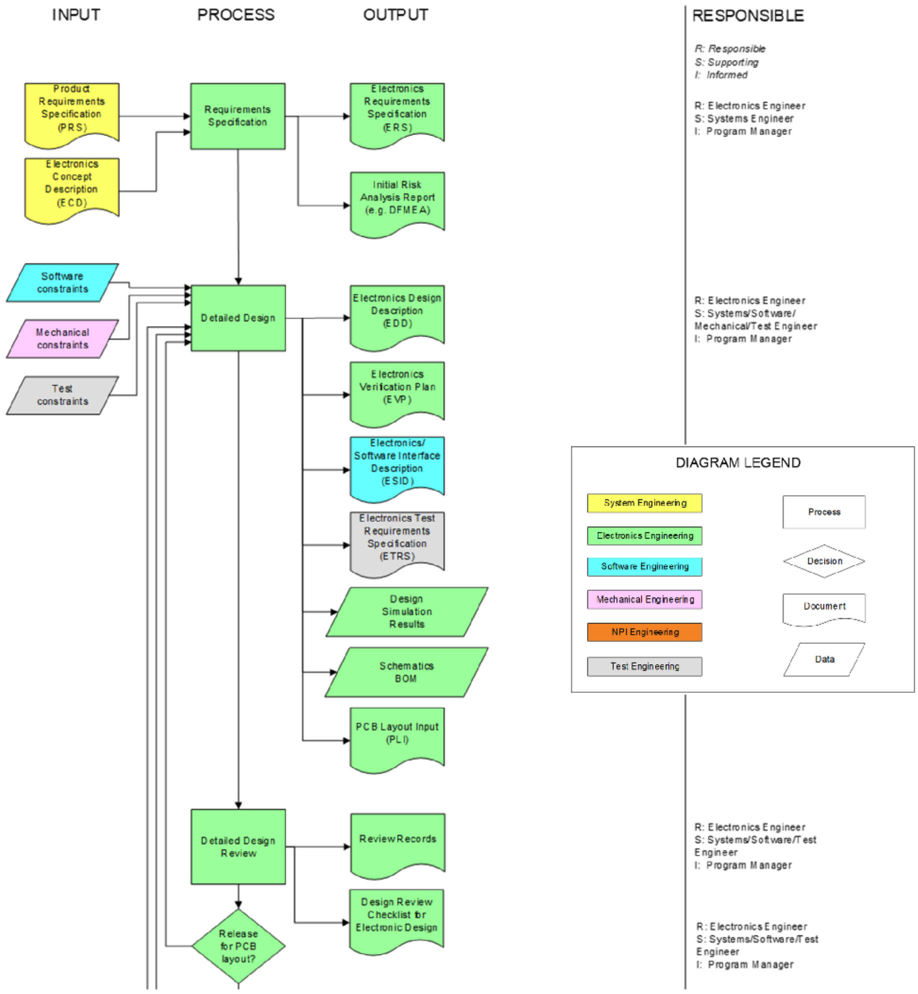


Figure 2.1 Electronics Engineering Process

Filename: AN-11020 Electronics Engineering Process.docx Page 8 of 27 Form: ANF-23003.doc Rev 7.0
Benchmark Confidential As covered by AN-23003
HARDCOPIES ARE CONTROLLED ONLY IF THE HARDCOPY CONTROL SECTION IS FILLED IN AND SIGNED.
OTHER HARDCOPIES ARE UNCONTROLLED. THE MASTER FILE IS AVAILABLE ON THE NETWORK.

Benchmark electronics

BEA Process Description
Electronics Engineering Process

QMS Document Number: AN-11020
Revision Level: 4.0

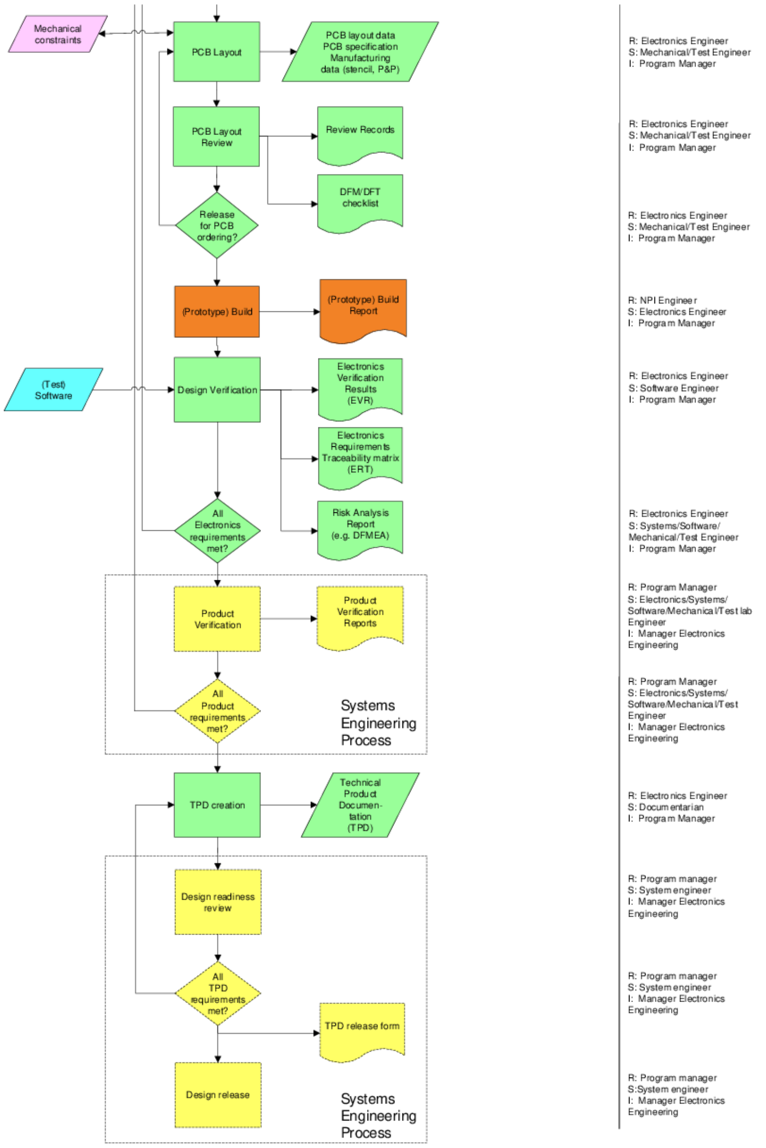


Figure 2.2 Electronics Engineering Process (continued)

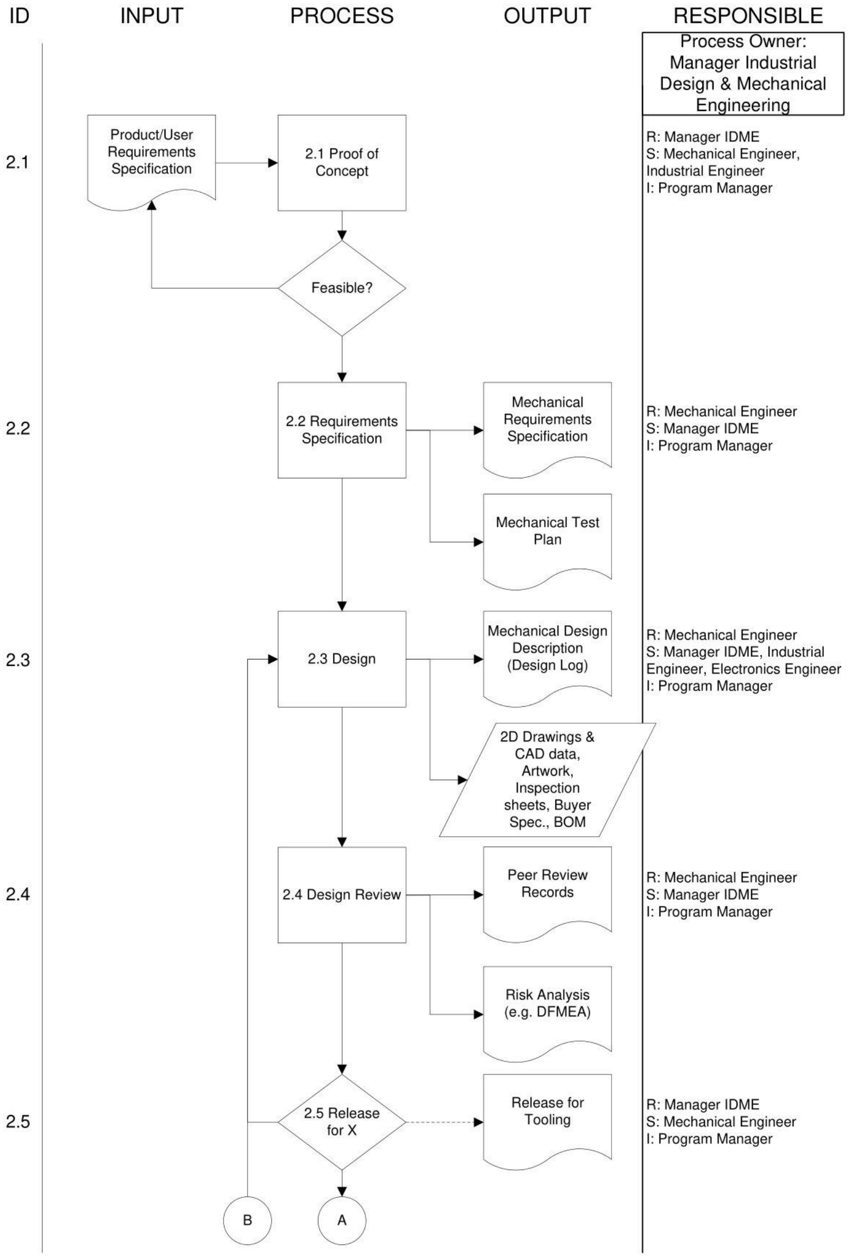
Filename: AN-11020 Electronics Engineering Process.docx Page 9 of 26 Form: ANF-23003.doc Rev 7.0
Benchmark Confidential As covered by AN-23003
HARDCOPIES ARE CONTROLLED ONLY IF THE HARDCOPY CONTROL SECTION IS FILLED IN AND SIGNED.
OTHER HARDCOPIES ARE UNCONTROLLED. THE MASTER FILE IS AVAILABLE ON THE NETWORK.

APPENDIX D — ME PROCESS

Benchmark electronics

Benchmark Almelo Procedure
Mechanical Engineering Process
QMS Document Number: AN-11015
Revision Level: 3.0

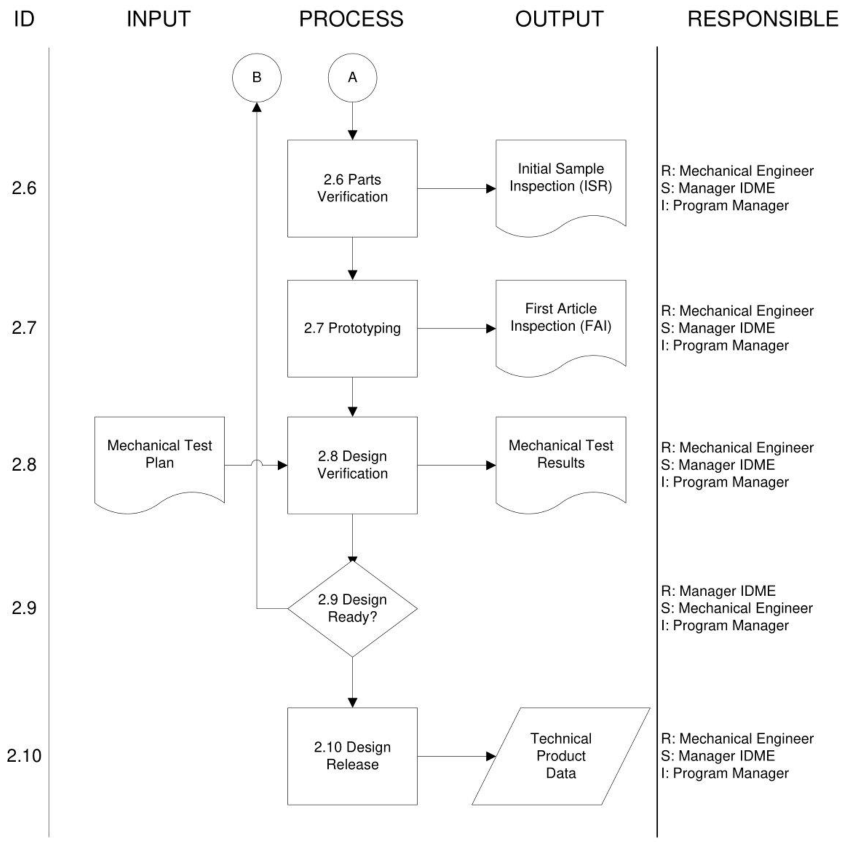
2 Mechanical Development Process



Filename: AN-11015 Mechanical Engineering Process.docx | Page 5 of 8 | Form: ANF-23003.doc | Rev 2.0
Benchmark Confidential | As covered by AN-23003
ALL COPIES ARE UNCONTROLLED. THE MASTER FILE IS AVAILABLE ON THE NETWORK.

Benchmark electronics

Benchmark Almelo Procedure
Mechanical Engineering Process
QMS Document Number: AN-11015
Revision Level: 3.0



Filename: AN-11015 Mechanical Engineering Process.docx   Page 6 of 8   Form: ANF-23003.doc   Rev 2.0
Benchmark Confidential   As covered by AN-23003
ALL COPIES ARE UNCONTROLLED. THE MASTER FILE IS AVAILABLE ON THE NETWORK.

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 <u>NOT</u> successfully completed projects of similar scope.													x	agree/disagree	1 = disagree, 4 = agree
C01	In the past, Benchmark has <u>NOT</u> successfully completed projects of similar scope.													x	agree/disagree	1 = disagree, 4 = agree
A10	The requirements supplied by the customer for this project are <u>NOT</u> well-defined.													x	agree/disagree	1 = disagree, 4 = agree
A11	In executing the project, the requirements supplied by the customer for this project have <u>NOT</u> 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 increases, 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 increases, 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 increases, 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 Work Breakdown Structure (WBS) that included task descriptions and work package descriptions.			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 participation 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 activities during reviews.										x				agree/disagree	1 = disagree, 4 = agree
K08	This project has a documented and practiced review process that examines completeness of configuration baselines.										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

Survey results

Respondent raw input

Question id.	WT_JAG score	MGO_SB score
A01	4	2
A02	3	2
A03	3	2
A04	2	3
A05	2	1
A06	2	1
A07	3	2
A09	2	3
C01	2	4
A10	2	1
A11	3	1
A12	1	4
A13	1	2
A14	4	1
B01		
B02		
B03	2	
B07		
B08		
B09	4	
O09	3	
B04		
B05		
B06	4	
N08		
O08	4	

Question id.	WT_JAG score	MGO_SB score
F01	4	3
F02	4	2
F03	3	2
F04		1
F05	3	1
F06	2	1
F07	2	2
F08	2	1
G01	4	2
G02	4	2
G03	3	2
G04	4	2
G05	3	2
G06	2	2
G08	3	1
G09	2	1
G10	4	1
G11	3	3
G12	2	2
G13	1	2
G14	4	4
O03	4	
O10	4	
H01	4	2
H02	3	2
H03	3	1

Question id.	WT_JAG score	MGO_SB score
D01	2	1
D02	2	3
D03	3	3
D04	2	3
D05	3	3
D06	4	2
D07	4	2
D08	3	2
D09	3	2
D10	3	2
D11	3	2
D12	3	2
D13	2	4
D14	3	2
D15	2	2
D16		4
D17	3	3
D18	3	1
D19	3	2
D20	3	2
D21	1	2
E01	4	2
E02	2	3
E03	1	1
E04	2	2
E05	2	1

Question id.	WT_JAG score	MGO_SB score
I01	4	2
I02	3	1
I03	3	2
I04	2	2
I05	4	2
J01	3	3
K01	4	4
K02	3	3
K03	2	3
K04	1	2
K05	2	1
K06	3	4
K07	4	2
K08	3	2
K09	3	3
L01	4	3
L02	4	3
M01	4	3
M02	3	4
M03	4	3
M04	2	3
N01	4	4
N02	3	1
N04		2
N05	2	1
O06	4	2
O07	4	2

** Blank cells were not entered by participants. This severely hindered the effectiveness of the survey,*

Results including weighted scores per assessment criteria

	Projects:	MGO_SB	WT_JAG
IPT	*integrated product teams*	1.80	2.20
PP	*project planning*	2.27	2.86
PMC	*management and control*	2.11	3.38
RSKM	*risk management*	1.63	2.88
REQ	*requirements management*	2.00	3.13
TRD	*trade-off studies*	1.67	3.33
ARCH	*architecting*	1.80	3.20
PI	*project integration*	3.00	3.00
VER	*verification*	2.67	2.78
VAL	*validation*	3.00	4.00
CM	*change management*	3.00	2.80
SEC_total		2.27	3.05
PC	*challenge*	2.07	2.44
	team size	8	9
	market	consumer	industrial
PERF	*performance*	#DIV/0! *	3.57

** Note that for* **MGO_SB** *the questions related to* ***PERF*** *were left blank by the respondent.*

APPENDIX D — INTERVIEW TRANSCRIPTS

PM of a small project in the industrial electronics domain (22-09-2022)

Summary

In this interview the PM discussed the roles and responsibilities of the different team members involved in their projects. They mentioned that the architect role that is responsible for the requirements and design is closest to that of the systems engineer as defined in this thesis (Section 3.3). The PM also discussed the importance of traceability and the use of tools such as Word, Excel, and Atlassian for tracking requirements. They emphasized the need to retain knowledge in long-running projects by keeping team members on board and maintaining thorough documentation. The PM also discussed the challenges of implementing automated testing and the use of compliance testing to ensure quality.

The PM in this interview discusses their role as being responsible for finances, contracts, and communication with the client. They also mention the importance of collaboration with the architect and the other team members involved in the project. Their management style is collaborative and client-focused.

As for their view on SE at Benchmark, the PM discusses the importance of traceability and maintaining thorough documentation. They also mention the use of tools such as Word, Excel, and Atlassian for tracking requirements and the challenges of implementing automated testing. This indicates that the PM values SE as an important part of ensuring the quality and success of projects.

The main points discussed in the interview:

1. **The roles and responsibilities of team members in a project:** The program manager discusses the different roles involved in a project, including the architect who is responsible for requirements and design, and the importance of collaboration among team members.
2. **The importance of traceability and thorough documentation:** The program manager emphasizes the need for traceability and maintaining thorough documentation in order to retain knowledge in long-running projects. They also discuss the use of tools for tracking requirements and the challenges of implementing automated testing.
3. **The value of collaboration and client-focused management:** The program manager discusses their role as being responsible for finances, contracts, and communication with the client. They also mention the importance of collaboration with the architect and other team members. This indicates a collaborative and client-focused approach to management.
4. **The role of Systems Engineering in ensuring project quality:** The program manager discusses the importance of traceability, thorough documentation, and the use of tools for tracking requirements, which are all key aspects of Systems Engineering. They also mention the challenges of implementing automated testing, indicating that Systems Engineering is an important part of ensuring the quality and success of projects.

Theme analysis (Dutch)

Themes	Excerpts
Project complexity	- Voor medisch en aerospace ben je wel verplicht een gevalideerde [requirements] tool te gebruiken. - Sommige projecten zitten in de maintenance fase: puur nog software releases. Hardware en mechanica staat al jaren vast. De klant wil dan soms een nieuwe software feature. Sommige

	projecten zijn nog in de definitiefase. Sommige projecten zijn net voor de releasefase.
View on Systems Engineering	- Ja, hij is verantwoordelijk voor de requirements en het design. Die stemt hij af met de klant. - [SE Decision Document] Lijkt me goed om bij te houden. Vaak met de klant worden dat soort keuzes ook gemaakt. Plusjes en minnetjes bijhouden inderdaad. - Aan de ene kant gaan we samen denken in een workshop en staan we echt in de frontlinie. Maar aan de andere kant heeft de klant echt een voordeel van expertise. - Hoe ik een SE voor me zie is als iemand die alle disciplines een beetje in toom houdt en zorgt dat het project niet een groot monster oid wordt. Wat je dan weg zag is dat de disciplines het beste voor zich kozen en dat het daardoor veel te complex werd. - Het moet met name een man zijn met erg veel ervaring. Die hoeft geen expert te zijn op elke discipline maar moet wel goed kunnen proeven wat nou echt het doel is van het project. Welk product moeten we nou maken, en dat in toom kan houden. - Op SE niveau is de klant echt verantwoordelijk en maken zij de keuzes.
Unforeseen changes/difficulties	- Het lastige is dat sommige oude projecten al tientallen jaren runnen dus niemand wil de migratie betalen. - Eigenlijk elke wijziging die je maakt moet een ticket voor komen. In TRAC. Ook alle mails en discussiepunten moeten worden opgeslagen. Dus voor alle wijzigingen is een logboekje te vinden met onderbouwing en informatie. - 9 van de 10 keer is dat [aanpassing van de planning] een verandering van de requirements.
Insight mistakes	- Je vertrouwt dan op de expertise van je team om ervoor te zorgen dat er niks stukgaat door een nieuwe release. - Het is heel veel common sense, niet alles is vastgelegd. We hebben echt vertrouwen in onze mensen wat dat betreft. Dat is wel typisch voor Benchmark.
Planning/management insight	- Program manager zorgt voor financiën en contracten en is het aanspreekpunt voor de klant. Die weet het beste wat de klant wil meestal. Dan hebben we een architect, die komt misschien het beste in de buurt bij de SE. Deze is niet bezig op implementatieniveau maar juist een hoger niveau, welke blokken het systeem opmaken. Junior en senior designer verzorgen de implementatie. We hebben nu ook een project lead / AKA senior engineer. - Ik vind het belangrijk dat de engineers de planning maken en dat ik kritische vragen stel.

Edited transcription (Dutch)

< Introductie van David over thesis >

Is SE een losse kostenpost geweest binnen de PROJECTEN X?

Nee. We kennen verschillende rollen: program manager zorgt voor financiën en contracten en is het aanspreekpunt voor de klant. Die weet het beste wat de klant wil meestal. Dan hebben we een architect, die komt misschien het beste in de buurt bij de SE. Deze is niet bezig op implementatieniveau maar juist een hoger niveau, welke blokken het systeem opmaken. Junior en senior designer verzorgen de implementatie. We hebben nu ook een project lead / AKA senior engineer.

Is deze architect dan ook verantwoordelijk voor de requirements en de tracability daarvan?

Ja, hij is verantwoordelijk voor de requirements en het design. Die stemt hij af met de klant.

Dan als we naar de implementatie gaan verliezen we een stukje traceability. We doen wel een compliance test: validatie. Bij een kleiner project, of echt iets nieuws dan ga je alles aftesten. Het meeste daarvan is handwerk.

Mits je al iets hebt staan, en je gaat dus voortbouwen op een release, dan testen we niet meer elke requirement apart, maar alleen wat er veranderd is; Alleen wat er nieuw is gebouwd. Je vertrouwt dan op de expertise van je team om ervoor te zorgen dat er niks stukgaat door een nieuwe release.

Het liefste heb je een automatisch systeem die dan duizenden requirements automatisch kan langsgaan. Echter dit kan louter voor software.

In wat voor tool tracken jullie de reqspec?

Dit gaat via Word en Excel. Voor industrial is er niet echt een standaard, dus wij definiëren in de DDP onze eigen tool.

Voor medisch en aerospace ben je wel verplicht een gevalideerde tool te gebruiken.

Liever gebruik ik een strikter systeem zoals Doors of Atlassian. Hoewel de migratie naar zo'n systeem lang duurt is het de moeite waard. Zeker voor automatische tests is dat super handig. Dan kan je 's nachts automatisch testen laten runnen op basis van je nieuwe functies van de dag ervoor.

Het lastige is dat sommige oude projecten al tientallen jaren runnen dus niemand wil de migratie betalen.

Bij zulke langlopende projecten, hoe waarborg je kennisbehoud in de organisatie?

Dat is inderdaad lastig. Je moet vooral de mensen aan boord houden.

Hoe zit het met de documentatie dan?

Eigenlijk elke wijziging die je maakt moet een ticket voor komen. In TRAC. Ook alle mails en discussiepunten moeten worden opgeslagen. Dus voor alle wijzigingen is een logboekje te vinden met onderbouwing en informatie.

Anderzijds voor elke regel code software wordt opgeslagen wie het heeft gemaakt en welk ticket erbij hoort.

Daarnaast maken we natuurlijk documentatie: requirements en design documentatie. Er wordt iets vergelijkbaars bijgehouden als een DHF.

Ook de review documentatie houden we bij en alles krijgt een netjes nummertje en wordt bijgehouden.

Het is heel veel common sense, niet alles is vastgelegd. We hebben echt vertrouwen in onze mensen wat dat betreft. Dat is wel typisch voor Benchmark.

Voor medisch en aerospace / meer kritieke projecten is strikter bijhouden wel wenselijk.

Wat denk je van het idee van een SE Decision Document?

Lijkt me goed om bij te houden. Vaak met de klant worden dat soort keuzes ook gemaakt. Plusjes en minnetjes bijhouden inderdaad.

Hoe zit het met de planning van het project?

Het is 1 klant met 13 projecten en 19 divisies.

Sommige projecten zitten in de maintenance fase: puur nog software releases. Hardware en mechanica staat al jaren vast. De klant wil dan soms een nieuwe software feature. Sommige projecten zijn nog in de definitiefase. Sommige projecten zijn net voor de releasefase.

Hoe pakken jullie die definitiefase aan? In hoeverre zijn de requirements dan al klaar?

Aan de ene kant gaan we samen denken in een workshop en staan we echt in de frontlinie. Maar aan de andere kant heeft de klant echt een voordeel van expertise.

Er is dan van bovenaf al vaak een roadmap van welke kant ze op willen. We kunnen wel adviseren maar qua strategie valt dat mee: wat algemenere dingen.

We hebben bijvoorbeeld meegewerkt aan een naval radio. Daarvoor waren de functionele requirements al helemaal duidelijk.

We zij n ook bezig met een oscilloscoop die allerlei metingen moet doen. Die metingen zijn al bekend en worden al jaren gedaan, dus ook daar zijn de functionele requirements al bekend.

Qua TRL is het dus vaak niet een hoog risico als ik het zo hoor?

Over het algemeen wel ja.

We hebben twee jaar geleden wel een meer definitiefase project gehad. Toen miste ik wel een SE vond ik.

Hoe ik een SE voor me zie is als iemand die alle disciplines een beetje in toom houdt en zorgt dat het project niet een groot monster oid wordt. Wat je dan weg zag is dat de disciplines het beste voor zich kozen en dat het daardoor veel te complex werd.

Het ging om een handheld apparaat. Hardware wilde behoorlijk wat elektronica en gooide het thermal probleem aan de mechanica mensen. Software wilde veel memory en processing power, maar elektronica vond dat weer niet fijn. Zo vecht iedereen voor haar eigen belangen en gaat het lang door.

Je mist dan dus een dedicated SE man, die dan een soort devils' advocate fungeert?

Het moet met name een man zijn met erg veel ervaring. Die hoeft geen expert te zijn op elke discipline maar moet wel goed kunnen proeven wat nou echt het doel is van het project. Welk product moeten we nou maken, en dat in toom kan houden.

Als je mensen los laat compleet dan loopt het stuk bij de integratie van het project.

Qua project management, hoe pak je het plannen aan?

Ik vind het belangrijk dat de engineers de planning maken en dat ik kritische vragen stel.

Via de lead engineers krijg ik de completion% door. Vooral financieel houd ik het in de gaten. De lead engineers zijn zo ervaren dat ik daar wel vertrouwen in heb.

De lead engineers hebben wekelijkse gesprekken met de klant en houden daar mooie verslagen van bij met de milestones, of die nog kloppen, actiepunten, etc.

Wanneer er dingen niet overeenkomen met de planning, wat is daar vaak de oorzaak van?

9 van de 10 keer is dat een verandering van de requirements.

In hoeverre is de klant dan echt deel van het ontwikkelteam?

Wij zijn juiste deel van het team van de klant. We werken met sprints van drie weken en dus elke drie weken wordt de backlog geupdate. En er is telkens ook een code release etc.

We houden de klant goed op de hoogte wekelijks en verder is het dus echt scrum.

Ik heb het idee dat deze projecten niet super SE zijn?

Klopt.

Mis je soms een SE?

Behalve het vorige voorbeeld niet echt.

Vaak is het ook niet onze verantwoordelijkheid. Het is echt aan de kant van de klant. Wij zijn echt toeleverancier van soft- en hardware. Soms worden er ook beslissingen gemaakt zonder ons erbij te betrekken. Dat is vervelend.

Op SE niveau is de klant echt verantwoordelijk en maken zij de keuzes.

Ben je het met me eens dat bij zo'n langer running project de SE louter belangrijk is in het begin in de definitiefase en dat de betrokkenheid dan snel afvlakt?

Ben ik met je eens, maar aan de andere kant denk ik dat wij de klant soms wel kunnen helpen met SE over de langere termijn. Dat is iets wat we meer zouden kunnen verkopen.

Voornamelijk voor de integratie van de verschillende disciplines en de keuzes die gemaakt worden zouden wij graag willen meedenken. Echter de klant staat het niet toe.

Het gaat niet vaak mis maar zouden wel graag willen meedenken meer. De vraag is hoe we de klant gaan overtuigen dat dat meerwaarde heeft.

<einde interview>

PM of Multiple small projects for a long-running client in the industrial electronics domain (22-09-2022)

Summary

The program manager is interviewed about a new product they are working on. The product is an upgrade of an existing one and the main challenge is implementing a new industrial ethernet standard (Profinet). The company lacked experience with this standard, so they hired a third party to handle it. However, communication with the third party was poor, causing delays and increased cost. The project ended up going over budget and changing from a fixed-price to a time and materials model. Furthermore, the project manager talks about their management style and how they try to stay involved in all aspects of a project.

The project manager in this interview has a hands-on approach to management, staying involved in all aspects of the project and having a good understanding of the technical details. They mention that they have a hardware background and thus some knowledge of electrical engineering, and they try to stay on top of everything that is happening in the project.

As for systems engineering, the manager mentions that Benchmark has a lot of experience with Linux and was able to estimate the effort required for the project. However, the new Profinet standard was completely new to them, so they had to hire a third party to handle it. This ended up causing delays and going over budget. The manager stresses the importance of good communication and collaboration with the client and other parties involved in the project.

The manager did not specifically mention the need for a dedicated Systems Engineer in this interview. However, they did talk about the importance of having a good understanding of the technical details of a project and the need for good communication and collaboration with the client and other parties involved. From this, it can be inferred that the manager values Systems Engineering and sees it as an important part of Benchmark's work.

The main points discussed in the interview:

1. **The challenges of implementing new technology:** The main focus of the interview is on the challenges that the company faced when implementing a new Profinet industrial ethernet standard in their product. The company did not have experience with this standard and had to hire a third party to handle it, which caused delays and increased the budget.
2. **The importance of good communication and collaboration:** The manager emphasizes the need for good communication and collaboration with the client and other parties involved in the project. They also mention that a lack of communication with the third party they hired caused delays and problems.
3. **The value of hands-on management:** The manager describes their approach to management as being hands-on and involved in all aspects of the project. They mention that they have a good understanding of the technical details and try to stay on top of everything that is happening.
4. **The role of Systems Engineering in the company:** The manager talks about the company's experience with Linux and their ability to estimate the effort required for the project. They also mention the importance of collaboration and communication, which are key aspects of Systems Engineering.

Theme analysis (Dutch)

Theme	Excerpts
Project complexity	- Nieuwe functionaliteit = connectiviteit en IOT. Dit maakte het een zeer complex nieuwe opdracht. - het Profinet protocol was compleet nieuw voor ons - de PRS was niet goed genoeg in het begin - We hadden de PRS echt beter kunnen leren kennen toen.
View on Systems Engineering	- Software man had een beetje SE kijk - een persoon die van bovenaf kijkt - Technisch inzicht - SE document waarin het concept etc. stond uitgelegd - Nu gebeurt dat [SE] impliciet. Iemand neemt die rol op zich, vanuit de belangrijkste discipline. - Making a concept description - Updating the PDD - Devil's advocate - optimisme drukken van jouw team en de klant een beetje kan drukken. Realistisch blijven. - SE moet er altijd op blijven. In het begin super druk, dan daarna houdt ie het team draaiende. Maar ook wanneer het product gelanceerd is en het team aan wat anders gaat blijft hij toch technisch verantwoordelijk
Unforeseen changes/difficulties	- verandering van fixed-price naar times-material - requirements moeten herdefiniëren. - requirements uit de klant erg globaal bleven. Wij hadden behoefte aan meer detail. Veel van die punten sleepten lang door. - Ik heb het idee dat het project bij de klant erg in prioriteit is gezakt.
Insight mistakes	- Even een thin-crossover schil. Dat duurde steeds langer om te maken - De klant wil wat en verwacht dat wanneer wij zeggen dat kan, dat wij daar ook verstand van hebben en dat dat zo gaat gebeuren - De impact van zo'n eenvoudig schilletje bleek niet goed ingeschat - Ik vraag me af als of we toen met de adviseurs opnieuw hadden gesproken dat we eerder hadden kunnen ontdekken dat zo'n schil te moeilijk had kunnen zijn. - te optimistisch. Te veel hooi op de vork. Uiteindelijk zei de senior architect maar laten we beginnen. Hij was behoorlijk pragmatisch.
Planning/management insight	- Een betere tooling waarin het allemaal kan in 1x zou mooier zijn. - In Jira dat plannen was erg ingewikkeld en onlogisch. En dus niet gekoppeld aan MS Projects.

Edited transcription (Dutch)

< Introductie van David over thesis >

Dit project is een vernieuwing van een oud product. Hoe zit het met de technologische uitdaging?

Er zijn gevallen van vernieuwingen waar de klant het systeemstuk zelf in handen heeft en anderen worden ingehuurd voor de details. Bijvoorbeeld een stuk elektronica in een vliegtuig waar onderdelen niet meer van te krijgen zijn, waar dus een vernieuwing van moet komen. Dan verandert er systeemtechnisch niks maar moet er wel wat gebeuren.

In dit geval is het echter anders: de concurrent snoepte steeds meer marktaandeel omdat zij een functie hadden die deze klant niet had. Dat was de Profinet standaard: industrieel ethernet welke beter is dan consumenten ethernet → zo goed als real-time (sneller dus) en betrouwbaarder. En beter gedefinieerd. Dit apparaat is een relay welke sensors en actuatoren verbindt met de PLC controller.

Verder wil de klant graag een soort nieuw OS erop met een web interface. Op afstand bekijken en software updates remote kunnen uitvoeren. Nieuwe functionaliteit = connectiviteit en IOT. Dit maakte het een zeer complex nieuwe opdracht.

Denk je dat we de kennis hiervoor al wel in huis hadden?

Dat idee had ik voor een deel, omdat het platform waar het OS op draait wel bekend is hier. We hebben heel veel producten die dit draaien met Linux. Onze mannen weten precies uit te rekenen hoe veel effort dat zou kosten.

Echter het Profinet protocol was compleet nieuw voor ons. Daarvoor zijn we op zoek gegaan naar een 3e partij die daar wel verstand van had.

We hadden dus een eigen platform van TI, en we zoeken een club die voor dat platform met Linux een stack kan leveren die dit regelt. We hebben een Duitse partij gevonden die dat kon regelen voor ons.

En wij dachten, met een goede software man van ons die een beetje System kijk had ook, moet dat goedkomen.

Ik voel al aankomen, dat ging dus niet goed?

Dat ging uiteindelijk inderdaad niet goed. Om te beginnen ging de club die wij hadden geselecteerd steeds niks van zich laten horen. Ze bleken een soort verlengstuk van TI te zijn, en ze moesten hun tijd besteden aan hun nieuwe chip. Wij kwamen daar te laat achter en waren dus zo 3-4 maanden verder.

Uiteindelijk hebben we toch een nieuwe club gevonden. En toen ging het goed.

Deze tegenval heeft ook gezorgd voor een grote budget vergroting en een verandering van fixed-price naar times-material.

Halverwege hebben we de klant bezocht, goed overlegd en uitgerekend welk deel van de fixed-price voor wiens rekening kwam.

Uiteindelijk ging het dus systeemtechnisch fout wat ons veel tijd en geld heeft gekost.

Hoe is jouw management aanpak? Hoe dicht zit je erboven op?

Ik ben van nature een hardware man en snap wel iets van EE. Maar van software weet ik niks. Dus bij de EE'ers kijk ik nog even over de schouders. Dat ging allemaal ook lekker. HW bordje was first-time-right voor 90%.

Onze software man was erg capabel dus op hem vertrouwde ik helemaal. Hij was erg verbaal en assertief en men was het met hem eens.

Wat we zagen is dat de partner op een ander software systeem (Ertos) draaide dan Linux. Volgens onze software architect was daar een eenvoudige oplossing voor. Even een thin-crossover schil. Dat duurde steeds langer om te maken.

De software partner wist ook zo snel niet hoe dat op te lossen. Uiteindelijk heeft de klant toegezegd toch maar geen Linux te gebruiken. Dan maar Ertos en geen breed toepasbaar OS (Linux). Hiervoor hebben we ook veel requirements moeten herdefiniëren.

Wanneer is deze fout gemaakt? Had het aan het begin kunnen worden opgevangen?

Nou zo'n klant (zeker als deze) kan je dat niet in de schoenen schuiven. De klant wil wat en verwacht dat wanneer wij zeggen dat kan, dat wij daar ook verstand van hebben en dat dat zo gaat gebeuren.

Die thin-crossover schil leek heel eenvoudig in het blok-diagram. Maar wat we leerden is dat, alles wat je in dat diagram toevoegt kost CPU power. En maakt de kans dat qua timing dingen mis gaan. → of de CPU trok het niet meer, of ergens was een pad te lang waardoor er timing issues ontstonden.

De impact van zo'n eenvoudig schilletje bleek niet goed ingeschat.

Was dit dan puur een gebrek aan communicatie tussen HW en SW? Of meer puur een SW fout?

CPU power is niet per se de software, maar het blijft een inschattingsfout. Ik vraag me af als of we toen met de adviseurs opnieuw hadden gesproken dat we eerder hadden kunnen ontdekken dat zo'n schil te moeilijk had kunnen zijn.

Maar aan de andere kant misten wij echt de kennis over Profinet dus waren we gewoon te optimistisch. En dan heb je ook nog een supplier van een stack die eigenlijk ook net te veel hooi op de vork had.

Hoe had je die ingeschat als risico?

Een aantal van de dingen waardoor we andere keuzes moesten maken stonden zeker weten boven de FMEA lijst, en zijn daar ook even gebleven.

Maar als je kijkt naar SE, hoe hebben we dat gedaan. Aan het begin geconstateerd welk platform → conceptual design. Gezien dat we een stuk mist dus dat er een 3e partij bij moest. Uiteindelijk zei de senior architect maar laten we beginnen. Hij was behoorlijk pragmatisch.

Maar eigenlijk was het beter geweest als we een persoon die van bovenaf kijkt moet over de hele looptijd van het project hadden gehouden. Nu wordt er even gekeken of het kan, gaat het project van start en krijgt die persoon het druk zat weer met andere dingen. Die zou moeten blijven monitoren zo van 'wat gebeurt daar?'.

Als Program Manager, had jij die persoon kunnen zijn? Valt die functie te combineren?

Dat is een hele lastige vraag. Combineer PM kunsten met technisch inzicht. Sommige kunnen dat, door hun verleden. Maar PMers worden niet afgerekend op technische inbreng zoals nu en moeten daar nieuw resources bij vragen.

Dan krijg je eigenlijk ook nog nieuwe extra projecten erbij. Drie of vier tegelijk.

Dus eigenlijk had iemand in het X team moeten zitten die deze rol vervulde?

Ja klopt. Dat was mooi geweest.

Nu gebeurt dat impliciet. Iemand neemt die rol op zich, vanuit de belangrijkste discipline.

Was SE nu wel los gebudgetteerd?

Ik denk het niet nee.

Met veel projecten is er wel een apart blad voor System werk, maar niet in dit project.

Je had het al over een blok-diagram. Was die en de info daarin voor iedereen beschikbaar?

Ja zeker. We hadden ook een SE document waarin het concept etc. stond uitgelegd. Die werd bijgehouden door onze software man. → Product Design Description.

Hoe veel veranderden de requirements over de loopduur van het project?

Afgezien van die grote tegenvaller zagen we wel dat de requirements uit de klant erg globaal bleven. Wij hadden behoefte aan meer detail. Veel van die punten sleepten lang door.

Had Benchmark daar meer de lead in kunnen nemen?

De klant vind dat zijn eigen verantwoordelijkheid. En eigenlijk vind ik dat ook, nu nog.

We zijn nu in de kwalificatiefase waarin het product wordt getest. Er zijn nu wat mankementen die te maken hebben met de context van dit product. Product is een subsysteem in een groter geheel, dus de cross-overs met de andere subsystemen in het grotere geheel en de requirements daarvoor zijn verantwoordelijkheid van de klant.

De klant heeft daar fouten gemaakt. En ik ben bang dat die niet met ons aan tafel gaan om die fouten op te lossen dus dat ze het maar gewoon zo gaan uitbrengen. Ik heb het idee dat het project bij de klant erg in prioriteit is gezakt.

Waarschijnlijk was die eerste requirements specificatie ook te nauwkeurig. Heel veel toepassingen zullen wel kunnen.

Klopt het dat heel veel van de fouten helemaal in het begin zijn gemaakt? De PRS was niet goed genoeg?

Klopt.

Hoe begon dit project? Was het een co-design sessie? Workshop?

Geen workshop. We zijn nooit bij elkaar geweest in het begin. Maar we hebben wel een paar keer super goede uitleg gehad van de product owner om het hele systeem te leren kennen.

Of een workshop toen had kunnen helpen denk ik inmiddels wel. We hadden de PRS echt beter kunnen leren kennen toen.

Gebruik je EVA voor het plannen van je projecten?

Nee

Hoe houd je dan inzicht op de completion status vs budget expended?

Je hebt een projectplan met alle stapjes erin die je hoopt te nemen. Ook met uren eraan gekoppeld. Projectplan was in Excel en ureninschatting in MS Projects.

Ik had wel goed zicht op waar we stonden en wat er gebeuren moest nog. Elke maand hadden we meetings waarin we dit uitrekende.

Een betere tooling waarin het allemaal kan in 1x zou mooier zijn.

We hebben ook Jira gepilot voor kanban. Waar ik mee zat is dat je prachtig sprints kan plannen uit de backlog. Maar ik wilde ook wel goed zien waar die sprints dan tot leiden. In Jira dat plannen was erg ingewikkeld en onlogisch. En dus niet gekoppeld aan MS Projects.

Was er documentatie over technische keuzes (decision-document)? Naast een CCB

Changes hebben we altijd goed bijgehouden. Alles boven de €1k kwam daar wel langs. We hebben toen in het begin die concept description gemaakt en die bleef eigenlijk wel. De PDD is ook wel eens geüpdatet.

Requirements management tooling?

Ook in Jira, via R4J.

Sluitende opmerkingen? Wat is nou achteraf jouw conclusie over SE?

Als je naar dit project kijkt dan is het optimisme in het begin onze downfall geweest. In het begin had er een echte SE man bij moeten zitten die moeilijke vragen stelt.

Je hebt iemand nodig die de devil’s advocate speelt en die het optimisme van jouw team en de klant een beetje kan drukken. Realistisch kan blijven.

SE moet er altijd op blijven. In het begin super druk, dan daarna houdt ie het team draaiende. Maar ook wanneer het product gelanceerd is en het team aan wat anders gaat blijft hij toch technisch verantwoordelijk.

<einde interview>

APPENDIX E — SE PROCESS DOCUMENT

Table of Contents

1 Document Information

1.1 Revision History

Revision	Author	Date	Changes
0.1	David de Groot	February 09, 2023	Initial revised version 2023 based on MSc thesis.
0.2	David de Groot	February 21, 2023	Update based on walkthrough comments from Henry Kompagnie.

1.2 Scope

Scope	Medical ISO 13485:2016	Avionics AS9100:2016	Industrial ISO 9001:2015
Compliant to	needs check	needs check	needs check
Compliancy verified on	xx-xx-xxxx	xx-xx-xxxx	xx-xx-xxxx

1.3 Purpose

The System Engineering (SE) process described in this document ranges from project conception to a released TPD of the device and finally its end-of-life. This document describes the overall responsibilities of the Systems Engineer: to ensure completeness, coherence, and integration of the solution and create a balance between its specification and execution.

The process describes the following:

- Deliverables created during the design process required for successful design, verification and validation, and acceptance by the customer.
- A process to produce and maintain these deliverables.
- Relations with the project management process and linked engineering processes (e.g. electrical, software, mechanical and test engineering, NPI).

The exact content and the amount of detail required for each deliverable is beyond the scope of this document. The process presented by this document is meant as a framework that is flexible enough to be adapted to all different types of projects and technologies.

1.4 References

The table below lists the Procedures and Forms that are referred to in this QMS document.

Reference	Document Title
BE-11003	Engineering Design Control Methodology rev. H
BE-11010	Design Review Procedure rev. F
AN-11015	Mechanical Engineering Process v. 3.0
AN-11020	Electronics Engineering Process v. 4.0
BE-11014	Software Engineering Process rev. E

1.5 Controlled Forms

The latest versions of the forms mentioned in the table below are approved through the approval of this document. However, updates of the forms below can be done without updating this procedure, as long as they still comply to it.

Doc ID	Document Title
-	-

1.6 Abbreviations and Definitions

Abbreviation	Description
BEA	Benchmark Electronics Almelo
BOM	Bill Of Materials
DFMEA	Design Failure Mode and Effect Analysis
DFM	Design For Manufacturing
DFT	Design For Test
ERS	Electronics Requirements Specification
ERT	Electronics Requirements Traceability matrix
ME	Mechanical Engineering
NPI	New Product Introduction
PCB	Printed Circuit Board
PM	Program Manager
PRS	Product Requirements Specification
P&P	Pick & Place
QMS	Quality Management System
SE	Systems Engineering
SEP	Systems Engineering Plan
SIS	System Interface Specification
SW	Software
TPD	Technical Product Documentation
TRL	Technological Readiness Level
TRS	Test Requirements Specification
VOC	Voice of Customer

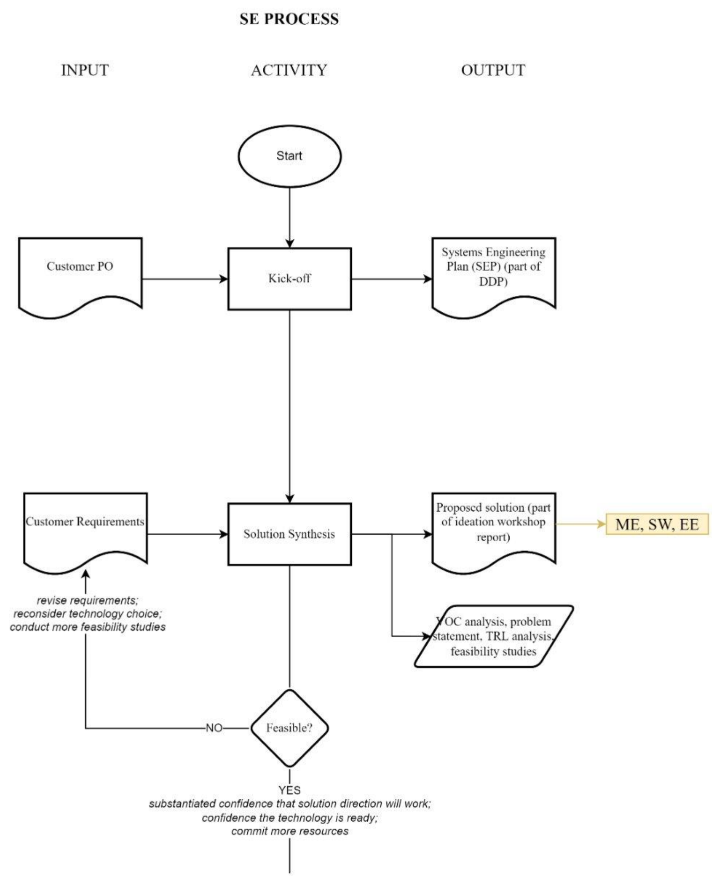
2 System Engineering Process

The System Engineering process enables the life cycle management of a system-under-design (SUD). For each stage of the system life cycle, the process enables an orderly progression through established decision-making gates to reduce risk and to ensure satisfactory progress.

- The process flowchart is shown in Section 2.1 Process flow
- Each phase of the flowchart is detailed in subparagraphs 2.1.0 to 2.1.9.
- Per deliverable, the scope, summary and reviewers are detailed in Section 2.2 Deliverables.

2.1 Process flow

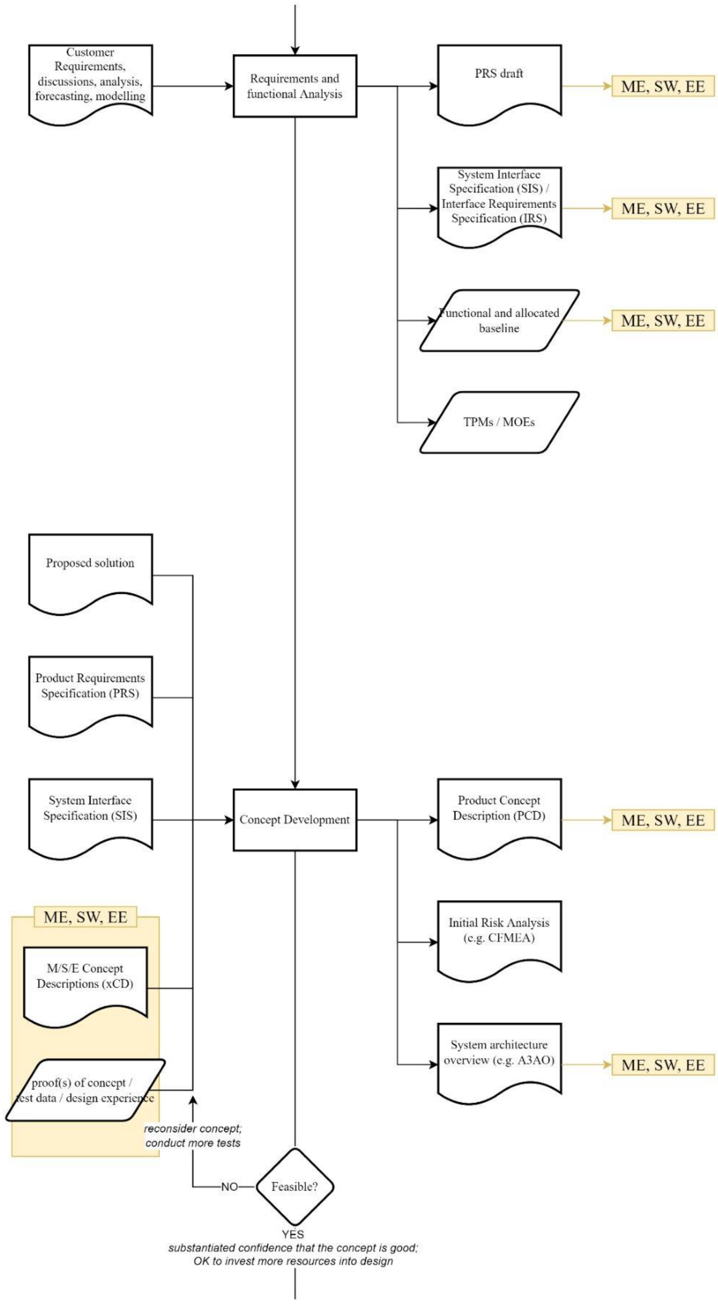
Figure 2.1 System Engineering Process shows the System Engineering process flowchart. In the following chapters the activities with their inputs, outputs and responsibilities are described in detail.



Continued on next page.

BEI Error! Unknown document property name.
System Engineering Process

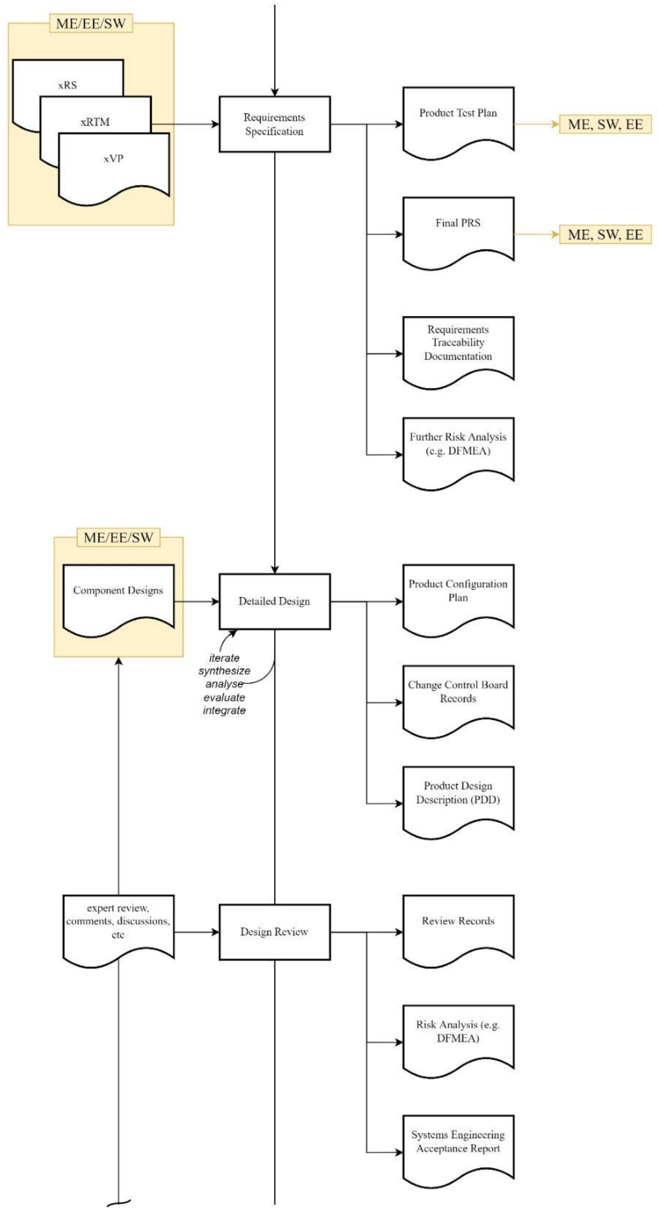
QMS Document Number: AN-11030
Revision Level: 2023-0.1



Continued on next page.

BEI Error! Unknown document property name.
System Engineering Process

QMS Document Number: AN-11030
Revision Level: 2023-0.1

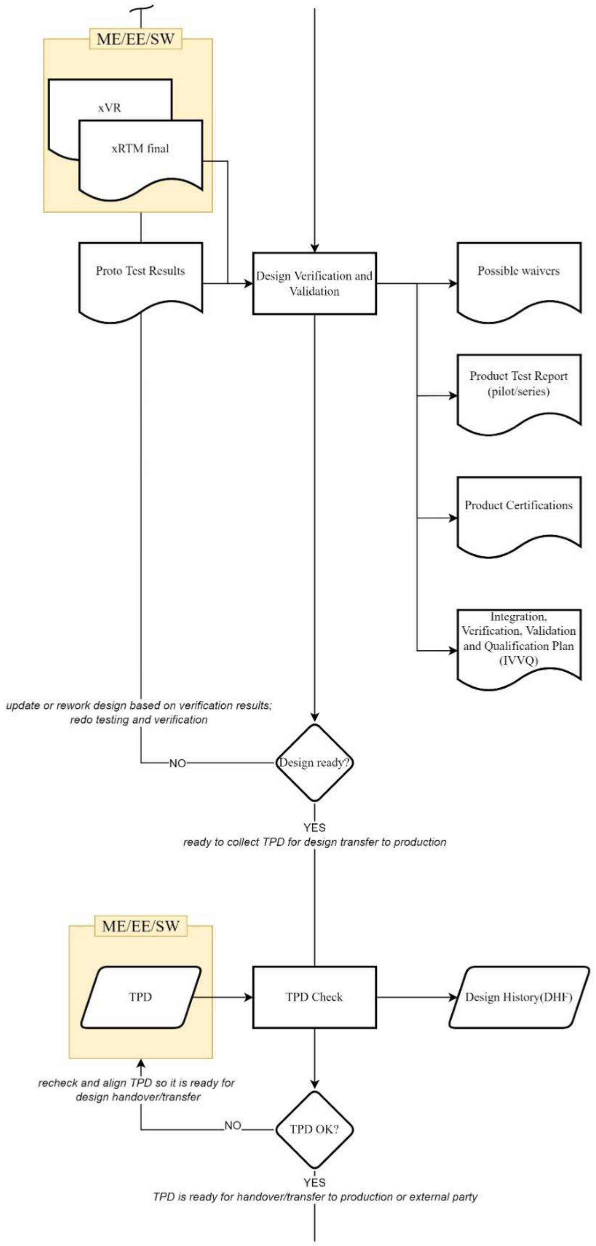


Continued on next page.

Benchmark electronics

BEI Error! Unknown document property name.
System Engineering Process

QMS Document Number: AN-11030
Revision Level: 2023-0.1



Continued on next page.

Filename: AN-11030_DG_v3.docx Page 8 of 25 Form: ANF-23003.doc Rev 7.0
Benchmark Confidential As covered by AN-23003
HARDCOPIES ARE CONTROLLED ONLY IF THE HARDCOPY CONTROL SECTION IS FILLED IN AND SIGNED.
OTHER HARDCOPIES ARE UNCONTROLLED. THE MASTER FILE IS AVAILABLE ON THE NETWORK.

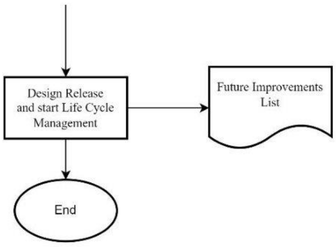


Figure 2.1 System Engineering Process

2.1.0 Kick-off / Proposal Phase

During this activity the systems engineer assists the PM in the kick-off of the project. The team is formed and initial budget and planning are assessed. Also in this activity, the systems engineer and PM should together determine the depth of SE applied in the project and records this in the SEP as part of the DDP. See Figure 2.2. Kick-off Phase.

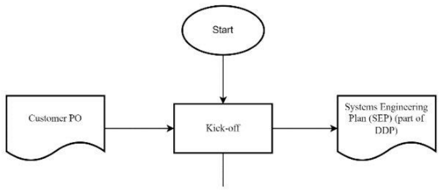


Figure 2.2. Kick-off Phase

Activities:

- Analyse customer business case
- Assess suitability for development project
- Initial budget and planning assessment
- Stakeholder analysis together with PM
    - Who develops the product?
    - Who builds the product?
    - What is the customer culture
- Assembling project team

Deliverables:

- Input for the quote of the Program Manager
    - Depth of SE activities
    - Project effort estimates (engineering)

2.1.1 Solution Synthesis Phase

In this phase the SE explores the problem space and creates an initial solution direction. As part of this activity, the VOC and system context are analysed. The SE determines a solution direction and conducts feasibility studies to investigate the feasibility of the solution and the TRL. The other disciplines may be involved in these activities as the SE sees fit. The proposed solution is used by the disciplines in their respective processes.

The system engineer delivers the proposed solution, based on early assessment of the customer requirements. See Figure 2.3. Solution Synthesis Phase.

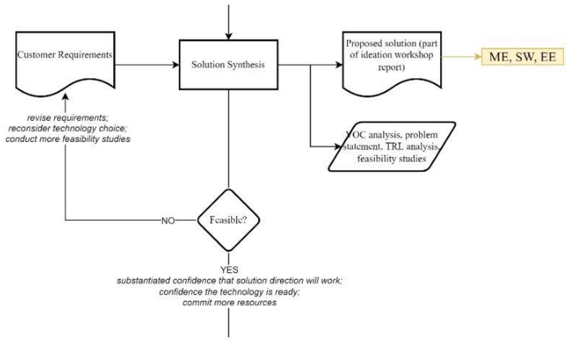


Figure 2.3. Solution Synthesis Phase

Activities:

- Analyse VOC
- Stakeholder analysis together with PM
  - Who develops the product?
  - Who builds the product?
  - What is the customer culture?
- Analyse system context, TRL
- Participating, leading in project scope discussions
- Determine solution direction or analyse existing concept
- Concept studies/Workshops
  - Research identified risk areas
  - Feasibility studies

Deliverables:

- Proposed Solution (and possible alternatives)
  - Problem Description
  - (Initial customer system requirements)
  - Product BOM estimates
- Project effort estimates (engineering)
- Results of analyses
  - TRL
  - Feasibility studies
  - Initial risk

2.1.2 Requirements and Functional Analysis Phase

In this phase the SE drafts a system level requirements specification, the PRS. The SE shall also collect any external requirements set by regulations or standards.The SE applies functional modelling and analysis to allocate functions to subsystems and assign responsibility to the subdisciplines.

Lastly TPMs are created, which drive design considerations. System interfaces are described either as part of the PRS or a separate SIS. The draft PRS and allocated baseline are used by the subdisciplines in creating discipline level requirements (and concepts) in the next phase. See Figure 2.4. Requirements and Functional Analysis.

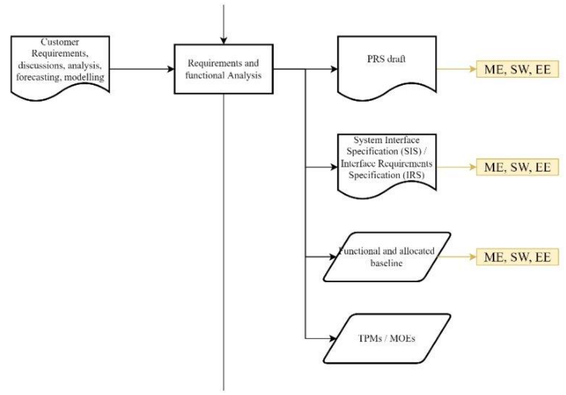


Figure 2.4. Requirements and Functional Analysis

Activities:

- Organising technical project meetings (CFD, Coaching for Deliverables)
- Perform System Requirements Engineering (Analysis, SMART, derived requirements, deviated requirements, feasibility/testability, what is NOT covered, ...) and draft PRS.
- Perform TPM analysis to find what technical performance measures are most important.
- Create functional and allocated baseline
    - Decompose functional requirements to domains (Software, Electronics, Mechanical)
- System Description Process (use case, scenario's, block diagram, data flow, behavioural, states, set of functions, failure modes, context diagram, technical risk analysis)
- Review of product solution with project team (PDR, Preliminary Design Review)
- Review of product solution with customer

Deliverables:

- Product Requirements Specification (PRS)
    - Functional Decomposition of architecture
    - Physical Decomposition of architecture (HW, SW, Mech)
    - Requirements Traceability Matrix (Caliber, Doors, etc.)
- System Interface Specification (SIS)
- List of TPMs/MOEs
- Product Design Description
- Product Test Plans

2.1.3 Concept Development Phase

In this phase the overall system concept is developed and approved in the form of a Product Concept Description (PCD).
As part of this, initial risk management is performed to inform the definition of the system concept. Additionally, as the concept is approved, a system architecture overview is created and from here on maintained to serve as single source-of-truth for the development team. Input to this activity are all previous analyses, test data, and design experience.
This activity works together closely with the disciplines since the discipline-level Concept Descriptions (E/S/MCDs) must match the PCD. See Figure 2.5. Concept Development Phase.

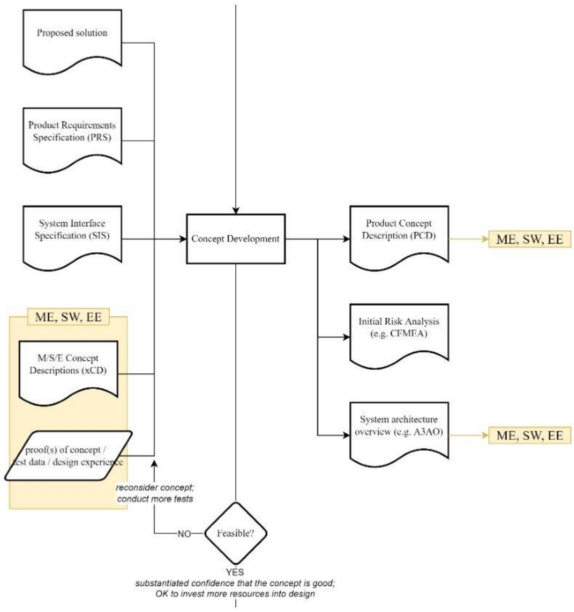


Figure 2.5. Concept Development Phase

Activities:

- Organising technical project meetings
- Create conceptual system designs
    - Document in Product Concept Description (PCD)
- Risk Management like CFMEA
- Create a System Architecture Overview
- Review of product solution with project team (Design Reviews)
- Review of product solution with customer

Deliverables:

- Product Concept Description (PCD)
- Updated Product Requirements Specification (PRS)
    - E.g. in Jira, Polarion, Doors, etc.
    - Including System Interface Specification (SIS) of Interface Requirements Specification (IRS)

2.1.4 Requirements Specification Phase

This Phase expands on the initial requirements and functional analysis, by combining it with the final approved concept, finalizing the PRS. Together with the final PRS, the system level Product Verification Plan (Test Plan) is created, along with requirements traceability documentation. Further risk management activities may also be performed. See Figure 2.6. Requirements Specification Phase.

Again this phase works closely together with the disciplines since the M/S/E Requirements Specifications, Traceability Matrixes, and Test/Verification Plans must match the System Level documents and 'talk to' each other.

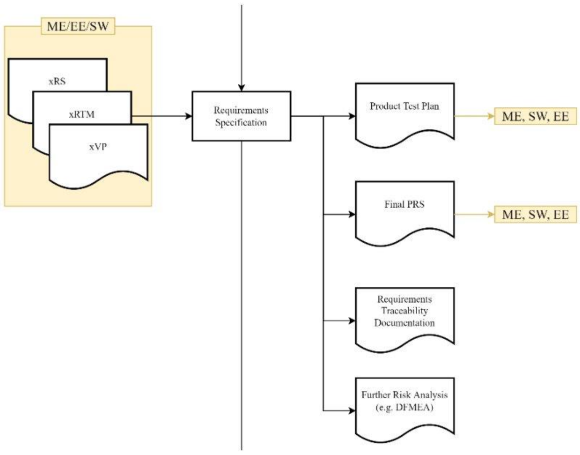


Figure 2.6. Requirements Specification Phase

Activities:

- Organising technical project meetings
- Finalizing PRS and Product Test Plan
- Assisting sub-disciplines with Requirements Traceability
- Update System Architecture Overview
- Further deeper level risk analysis such as DFMEA

Deliverables:

- Updated architecture overview
- Updated and final Product Requirements Specification (PRS)
    - Including System Interface Specification (SIS) of Interface Requirements Specification (IRS)
    - Requirements Traceability down to disciplines (xRTMs)
- Initial System level testing procedure or master plan (PTP, Product Test Plan)

2.1.5 Detailed Design Phase

This activity is cyclic, as the system is detailed more in-depth and as part of this activity the system implementation will progress through several baselines. For this a proper Configuration Plan must be established. Any changes made to the system must pass through the CCB, of which the systems engineer is a part.

The main tasks of the systems engineer in the Detailed Design phase are documenting design choices and assisting the disciplines with trade-off studies, taking into account input from the PCD, PRS, and constraints from the disciplines. See Figure 2.7. Detailed Design Phase.

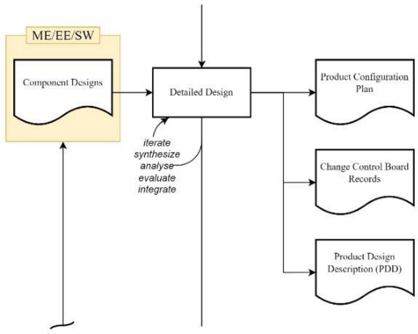


Figure 2.7. Detailed Design Phase

Activities:

- Organising and Leading Design Review meetings and DFMEAs (BE-11018)
- Coach the design processes (HW, SW, ME, …) by reviewing of the designs, initial architecture with customer requirements in mind
- Guarding consistency of architecture and design for integration and testability
- Further risk analysis (e.g. DFMEA)
- Planning of Product Configurations

Deliverables:

- Review Records of Design Reviews
- Product Test Procedure
- Product Configuration Plan (Proto A, Proto B, Pilot, etc.)
- System Engineering Acceptance Report
  - (Requirements coverage, successful design reviews, DFM/DFT, Engineering changes implemented/open, design risk management etc.)
- Design Failure Mode and Effect Analysis (DFMEA)

2.1.6 Design Review Phase

Design reviews are performed at the end of each project phase, or intermittently at appropriate project Design reviews are conducted at the end of each project phase or at key milestones to systematically assess design results and provide feedback to designers on existing or emerging problems.

The systems engineer is a key member of the review team and the outputs include Review Records, risk analysis documents, and an SE Acceptance Report (SEAR) that addresses and resolves technical issues. The SE Acceptance Report outlines the cause and actions for each issue, with the goal of prevention, correction, or improvement, and is maintained throughout the project life cycle as part of a broader SE Acceptance and Control process, which involves reviewing and prioritizing potential action plans and analysing interrelated issues for a consistent and cost-effective resolution. See Figure 2.8. Design Review Phase.

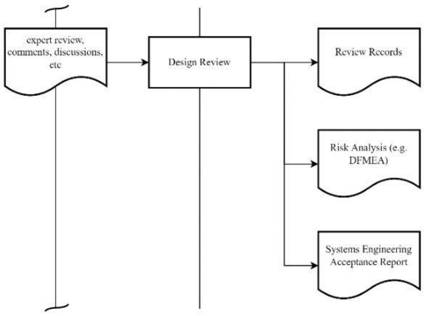


Figure 2.8. Design Review Phase

Activities:

- Critical Design Review
- Manage Design Changes (Change Control Board)
- Further risk analysis (e.g. DFMEA)
- Set the Product Acceptance Criteria

Deliverables:

- Review Records of Design Reviews
- System Engineering Acceptance Report (SEAR)
  - (Requirements coverage, successful design reviews, DFM/DFT, Engineering changes implemented/open, design risk management etc.)
- Change Control Board Records
- Design Failure Mode and Effect Analysis (DFMEA)

2.1.7 Design Verification and Validation Phase

The System Verification and Validation activity (Figure 2.9. Design Verification and Validation Phase) ensures that the designed system meets specified requirements and is conducted throughout the system's life cycle, including during development. The SE is responsible for verifying that the system level design matches requirements and assists in discipline-level verification activities. The verification process should be applied to all activities and products produced during the development process to ensure the system meets necessary requirements and functions as intended both during development and throughout its lifetime.

During this phase, the majority of the project activities are supporting the engineering processes (mechanical, software, electrical, etc.).

BEI Error! Unknown document property name.
System Engineering Process

QMS Document Number: AN-11030
Revision Level: 2023-0.1

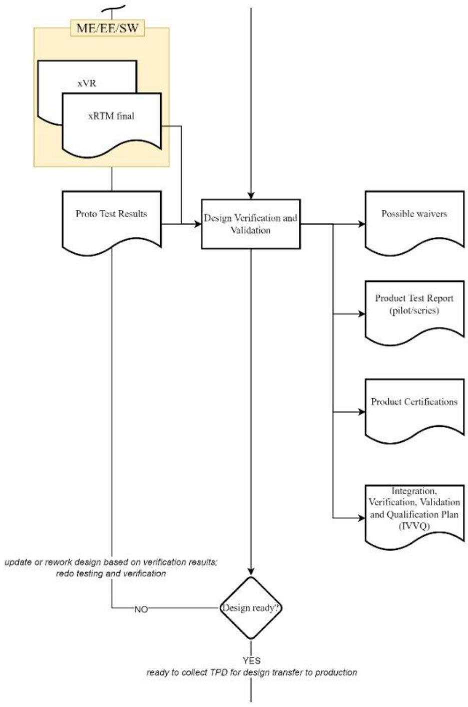


Figure 2.9. Design Verification and Validation Phase

Activities:

- Lead and coordinate integration of the product and its verification
- Transfer the Integrated / Verified System to the Customer
- Validate the performance and characteristics of the system w.r.t. stakeholder requirements
- Show Requirements Coverage (up to 100% ?)
- Discuss non-compliances, contractual obligations with the customer, including waivers

Deliverables:

- Product Test Report
    - Integration Test Results
    - Accomplishments/Acceptance Summary
- Update of Requirements Traceability Matrix
- Update of Technical Risk Analysis
- Possible Waivers
- Product Certifications

2.1.8 TPD check Phase

This activity occurs in the final stage of the project, before the Design Release and transfer to NPI Engineering. While it is the responsibility of the TPM to collect and properly archive the DHF, the systems engineer has an important role in ensuring technical comprehensiveness of the TPD that will be archived in the DHF.

Filename: AN-11030_DG_v3.docx Page 17 of 25 Form: ANF-23003.doc Rev 7.0
Benchmark Confidential As covered by AN-23003
HARDCOPIES ARE CONTROLLED ONLY IF THE HARDCOPY CONTROL SECTION IS FILLED IN AND SIGNED.
OTHER HARDCOPIES ARE UNCONTROLLED. THE MASTER FILE IS AVAILABLE ON THE NETWORK.

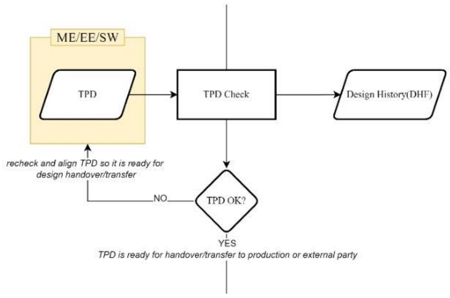


Figure 2.10. TPD check Phase

Activities:

- Ensure technical comprehensiveness, completeness, unambiguity of the TPD
- Coordinate design handover activities

Deliverables:

- Product Certifications
- Delivery of Design History File

2.1.9 Design Release / Start Life Cycle Management

In project closure phase, the systems engineer participates in the change control board and keeps track of the future improvements. In this activity the design is released for production, and the project is concluded for Design Engineering. The systems engineer and TPM have a role here in performing a retrospective to assess the technical considerations made and the applied processes. Future improvements should be noted and made readily available as design input for new projects.

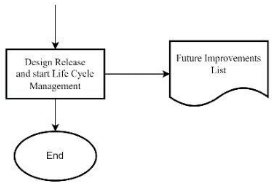


Figure 2.11. Design release and life cycle management phase

Activities:

- Agreements on product life cycle management
- Transfer to sustaining
- CCB participation
- Identify problem areas, future improvements, through retrospectives

Deliverables:

- Future Improvements List
- Change Control Board Records

2.2 Deliverables

2.2.1 Deliverable: Systems Engineering Plan (SEP)

Scope

The SEP is part of the DDP: the overall controlling set of documents that describe the project/program. The PM parts of the DDP mainly focus on the program structure, responsibilities and activities. As well as personnel and control management. The SEP parts focus on the technical project planning and control. It includes a plan on how to perform the requirement management activities in the project. The SEP is the result of the concept phase in which the (technical) project and associated WBS is defined. The SEP is further used as input for subsequent development plans for SW, HW and ME.

Summary

The SEP addresses the strategy and approach for all technical aspects of the project, including the system architecture and development, risk management, quality management, configuration management, information management, verification and testing, integration, validation, production, and deployment. SE planning should account for the full scope of technical activities and ensure that they are integrated in order to achieve a comprehensive and integrated plan for the project.

This document is used to tailor the various activities and SE process to the needs of the project/program and is used to control the systems development when completed and approved. The SEP further defines the specifications and documentation to be written (DHF), the baselines, requirement management and configuration management. The Configuration management plan and the Requirements management plan may be, dependent of the size of the project contained in the SEP.

The Requirements Management Plan (RMP) chapter describes all activities that need to be performed for requirement management. Requirement management includes the tools and processes that are followed for requirement traceability, requirements verification and cross referencing. The requirement management process is a support process and can be tailored for specific project needs. Tailoring of the process is described in this plan and should be approved by all stakeholders and reviewers.

If there is no general requirement management process, this plan is leading for the RM activities. If there is a general RM process, this plan only describes the differences to the main process.

Reviewers

It is reviewed by a team including the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

2.2.2 Deliverable: Proposed solution

Scope

The Proposed Solution shall comprise general subsystem lay-out and the initial direction for the technological solution to the identified problem. Depending on the complexity of the project it may also include a VOC/CTQ analysis, TRL analysis, and initial feasibility studies to support the TRL verdict. The level-of-detail for this document is low: it should be enough to allow the creation of an initial schedule and budget, but it may change in the concept development phase of the project.

Summary

As part of the *Solution Synthesis* activity, a solution direction is determined. This direction (or directions) shall be documented in a Proposed Solution document.

Reviewers

It is reviewed by the PM, systems engineer, relevant discipline lead engineers, and the customer product manager and other representatives like marketing, and sales.

2.2.3 Deliverable: Product Requirements Specification (PRS)

Scope

The PRS is the result of the translation of the customer requirements. Is used as input for the applicable disciplinary (SW/HW/MECH) requirements specification. This document is intended to explicitly describe what the product should do.

Summary

The PRS is a document that lists all product (system) level requirements and serves as the result of the translation of customer requirements. It is used as input for the applicable disciplinary (EE, SW, and ME) Requirements Specifications, and it is intended to explicitly describe what the product should do.

The requirements described in the PRS serve as an agreement with the customer and are divided into two groups: Functional and Non-Functional requirements. The Functional requirements are expressed as use cases and scenarios involving certain actors in the different subsystems, while the Non-Functional requirements relate to things like performance, environment, standards, and usability requirements.

The PRS shall specify what the system shall do, but not how it should be done. For example, it may specify the system's performance in terms of the number of operations per second, but it will not specify which microprocessor should be used to achieve this performance. The system will be validated during and at the end of the design process based on the PRS, for example by testing whether the actual number of operations per second matches the specification in the PRS.

Reviewers

The PRS is reviewed by a team including the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

2.2.4 Deliverable: System Interface Specification (SIS) / Interface Requirements Specification (IRS)

Scope

The SIS is output of the concept phase and may be updated during detailed design phase. For some complex projects it can be beneficial to create a separate requirements category and document for the requirements for interfaces between subsystems, and for the interfaces between the system and its environment.

Summary

The purpose of the system interface specification is to provide a definition of the product interfaces (external and internal). This includes user, electrical, mechanical, software, optical, thermal and other interface parameters and connections. The PM and systems engineer determine at the start of the project whether this shall be a part of the PRS or a separate document like the SIS or IRS.

Reviewers

It is reviewed by the PM, systems engineer, EE and SW Architects and lead engineers, and the customer product manager.

2.2.5 Deliverable: Product Concept Description (PCD)

Scope

The PCD is the output of the Concept Phase. It redefines the initial Solution Direction into a more refined and final concept to enter the Design Phase with. The PCD is approved by the design team and the customer and finalized/locked when the project exits the Concept Phase. The PCD is input to the PDD, which is created and maintained over the Design Phase.

Summary

The PCD is a system-level document serving as the output of the Concept Phase. It includes all design considerations that went into concept design and choice, including trade-off studies, proofs-of-concepts, prototypes and tests to verify the concept is OK to continue the project with. It should be of sufficient level of depth that it offers the customer a substantiated view that the concept will fulfil its goal, but without yet entering the design phase and thus with minimal 'design' effort.

Reviewers

The PCD is reviewed by the PM, systems engineer, relevant discipline lead engineers, and the customer.

2.2.6 Deliverable: Failure Mode Effect Analysis (xFMEA)

Scope

In the design process risk management activities are performed at various stages and levels (e.g. CFMEA, DFMEA, FMEA). The systems engineer is responsible for the creation and maintenance of risk management documentation over the system life cycle. There should be compliance to BE-11008.

The scope of these FMEAs is limited to the concept and the design phase of the project. Potential manufacturing issues should be transferred to the Process FMEA.

Note: *For redesigns, the DFMEA is repeated.*

Summary

An FMEA is tool for risk management on the product level. The intention of an FMEA is to list all the functions of the product and analyse potential single cause single fault conditions that may affect the performance of the device. The FMEA may be repeated at various stages and levels in the design process (e.g. Concept FMEA, Design FMEA, Process FMEA)

Reviewers

The various risk analysis documents are reviewed by a team consisting of the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

2.2.7 Deliverable: System Architecture Overview (AO)

Scope

The goal of the AO document is to present an overview of the system architecture that can be used as a starting point for learning and a way to communicate the system architecture with stakeholders. It should not replace existing documentation, although in some cases it may, but instead present a focused overview of the current standing of the system design.

Summary

In complex development efforts it is difficult to keep the whole team and everybody involved up-to-date with the current state of system design. This is not because design considerations are poorly documented, but because the amount of information in the documentation is often very extensive (as it should be), but therefore long, hard to read, and sometimes outdated (or WIP). A System AO document overcomes this problem by providing an understandable, and accessible for all, single source of truth that is updated throughout system design.

One such way is via an A3 AO[1], although the SE is free to decide the exact form of this deliverable. The goal should be to create an easy to understand document that can be checked quickly and serves as a good basis for understanding the system design.

[1] A3 Architecture Overview: https://a3ao.eu/

Reviewers

The AO is continually reviewed because it is in use by the design team (PM, systems engineer, engineers) and the customer.

2.2.8 Deliverable: Product Design Description (PDD)

Scope

The PDD is an evolved version of the PCD. It defines the system at a higher level of detail. Included in this document should be the System Definition with Operational Requirements (Need, Mission, Use Profile, Life Cycle), Maintenance Concept, Functional Analysis and Allocation, Interface Specification, Performance Measures, Physical and Usability (Human Factors) Characteristics, and Sustainability and End-of-Life Characteristics.

The document shall also describe Design and Construction characteristics, including CAD/CAM Requirements, Materials, Processes and Parts (BOM), EMC, Safety, and Testing Considerations, and data on Economic Feasibility. Furthermore, included should be Logistics information such as Maintenance Requirements, Supply-Chain, Personnel and Training, Facilities and Equipment, Quality Control, Packaging, Handling, Storage, and Transportation Requirements, and Customer Service Considerations.

Summary

The PDD is an evolved version of the PCD. It defines the system at a higher level of detail.

Reviewers

It is reviewed by the PM, systems engineer, EE and SW architects and lead engineers, and the customer product manager and other representatives like marketing, and sales.

2.2.9 Deliverable: Product Test Plan (PTP) / Design Verification Plan (DVP)

Scope

The PTP or DVP documents the product-level testing to be performed for verification of the system. The plan should describe how all requirements are to be verified, in what time frame, and under what external conditions, including specifics like test equipment and settings. It should integrate with the discipline-specific test plans.

Summary

The PTP or DVP documents how testing should be performed for verification of the system. All requirements shall be tested, or else a waiver provided. Based on the PRS, the PTP describes how it is ensured (e.g. analysis, test, inspection, demonstration) that the product requirements are met. Required test provisions shall be determined in this document.

Reviewers

The PTP/DVP is reviewed by the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

2.2.10 Deliverable: Product Configuration Plan

Scope

The product configuration plan records the desired configuration(s) at early concept phase (e.g. proto A, proto, B, pilot, or more), and existing product configurations during subsequent phases in the product life cycle.

Summary

The Product Configuration Plan document is initiated during the concept phase of the product's development and updated throughout its life cycle. It records the desired configuration(s) of the product at the early concept phase and any existing configurations during subsequent phases. This plan serves as a reference framework for product configurations and variants, and it may be used as input for the development of electronics, mechanics, and software architectures, as well as the Product Test Plan and factory test requirements.

In addition, the Product Configuration Plan helps the project manager to maintain the development scope and ensure that major product variants are identified and included in the proposed solution at the proposal phase.

Reviewers

The Product Configuration Plan is reviewed by a team of professionals including systems engineer, EE and SW Architects and lead engineers, NPI/setup and test engineering, and the customer product manager and other representatives like marketing, and sales.

2.2.11 Deliverable: Change Control Board Records (CCB Records)

Scope

The CCB is a group that is initiated after the project enters Change Control. Change Control is a process that is initiated after major design verifications are started to prevent the need for these verifications to be repeated, which can cause budget constraints. The CCB may continue to operate as long as required by the program management.

Summary

Change Control Board Records contain a summary of discussed items, decisions made and open action items.

Reviewers

The CCB Records contain a summary of the items that have been discussed, the decisions that have been made, and any open action items. The meeting minutes from the CCB are reviewed by all members of the CCB.

2.2.12 Deliverable: Systems Engineering Acceptance Report (SEAR)

Scope

The SEAR provides the overview of the compliancy of the **design** (before realization).

Summary

The SEAR is a document that provides an overview of the compliance of the design (before it is realized). It identifies the compliance of the design with customer requirements and may be accompanied by an updated customer requirements matrix. The focus of the SEAR is the assessment of whether or not the design meets the required compliance.

If the compliance itself cannot be provided yet (e.g. due to the design's EMC behaviour), then a risk assessment of those requirements is included in the SEAR. The document may also include a description of the design's reusability.

Reviewers

The SEAR is reviewed by a team including the quality manager or project quality engineer, PM, systems engineer, relevant discipline lead engineers, and customer representatives.

2.2.13 Deliverable: Product Test Report (PTR)

Scope

The Product Test Report is a document that gathers all relevant information after the Realization Phase (e.g. Proto A/B) and the Verification and Qualification phases (e.g. Pilot/Series). It contains all test results and summarizes the configuration baseline, open issues and problem report identifications, agreements on incompliances, and the coverage of customer requirements.

Summary

The product test report summarizes:

- Configuration Baseline.
- Open Issues / Problem Report identification / Agreements on incompliances
- Coverage of customer requirements (See also “Update of Requirements Traceability Matrix”).
- Declarations of performances (e.g. test house test reports, CE-marking description, ...)

Reviewers

It is reviewed by all involved discipline lead engineers, members of the CCB, and the customer.

2.2.14 Deliverable: Possible Waivers (ANF-23010)

Scope

Waivers are written to obtain formal approval from the customer for deviations in the design process. Waivers are used when a project plans to deviate from agreements or requirements at any stage, including the requirements verification phase and the pre-production/pilot series. A waiver is a request for formal approval of a deviation that has already occurred, while a deviation request is a request for approval of a deviation that is planned for the future.

Definitions: In case the deviation has already taken place it is a “Waiver” request and in case the plan is to deviate in the future it is a “Deviation” request.

Summary

A waiver is written for the purpose of registering and obtaining formal (customer) approval for deviations.

Reviewers

Originator, (Project) Quality Engineer, Quality Manager, Program Manager, Customer

2.2.15 Deliverable: Product Certifications

Scope

This deliverable summarizes declarations of performances such as external Test Reports and CE-marking descriptions and demonstrations of the fulfilment of the various quality assurance and performance tests.

Summary

Certifications are formal proof that the product complies with the required regulations or stipulations present within the contract with the customer or imposed by an independent, national, or international organization.

Reviewers

Originator, (Project) Quality Engineer, Quality Manager, Program Manager, Customer

2.2.16 Deliverable: Integration, Verification, Validation, and Qualification Plan (IVVQ)

Scope

The IVVQ defines the final integration and verification strategies for the product. It is based on input from the SEP and DDP, as well as the conceptual design and requirements specification documents.

Summary

The integration part of the plan describes how the different system components (coming from different disciplines) will be put together and tested.

The verification part of the plan describes how the design will be verified, checking compliance with all requirements. The plan should describe how all requirements are to be verified, in what time frame, and under what external conditions. This is an evolved and final version of the PTP/DVP.

The validation part of the plan describes how the system performance will be validated to ensure that it meets the customer's expectations and requirements.

The qualification part of the plan outlines the qualification tests that need to be done on the design to meet regulatory requirements, and it also allocates resources, time, and locations (which may be external) for these tests.

Reviewers

The IVVQ is reviewed by a team including the quality manager or project quality engineer, PM, systems engineering, relevant discipline lead engineers, and customer representatives.

2.2.17 Deliverable: Future improvements list

Scope

The future improvements list documents the results of internal retrospective sessions, feedback from the customer or end users, changes in technology or market trends, or new features that were not included in the initial release.

Summary

As the product enters its life-cycle management phase after production, the systems engineer shall stay involved to monitor and index possibilities for future improvements. This is because after release, it is still possible for the product to be updated and improved. These improvements could be based on internal retrospective sessions, feedback from the customer or end users, changes in technology or market trends, or new features that were not included in the initial release.

By maintaining this list of potential improvements, the systems engineer can ensure that the product stays current and relevant over time. Additionally, this list can be used to prioritize and schedule future updates to the product, and to track the progress of these updates. It can help the systems engineer to identify any potential issues or challenges that may arise during the development of these updates, and to plan for how to address them. Finally, this list may also be useful during the sales trajectory for acquiring new projects.

Reviewers

It is reviewed by a team including the quality manager or project quality engineer, PM, systems engineering, relevant discipline lead engineers, and customer representatives.

End of document