Technische Universität München
TUM School of Engineering and Design

Product Family Design: an approach to maintain external variety, manage product complexity, and minimize cost

Sebastian Josef Rötzer

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Karsten Stahl

Prüfer*innen der Dissertation:

1. Prof. Dr. Markus Zimmermann
2. Prof. Dr.-Ing. Dieter Krause

Die Dissertation wurde am 22.09.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 30.01.2023 angenommen.

"se hace camino al andar."[1]

Campos de Castilla, Proverbios y cantares, XXIX
Antonio Machado (1875 – 1939)

[1] der Weg ensteht im Gehen.

Ich danke all jenen aus tiefstem Herzen, die mich auf meinem Weg unterstützt haben.

Sebastian Rötzer

TABLE OF CONTENTS

Abstract

Companies offer wide portfolios of their products on the market to satisfy various customer segments and to achieve high sales volumes. By doing so, the external variety of products increases. Unfortunately, this often comes along with an increasing internal variety of components. As products become more complex, companies face more difficulties in identifying synergies within their product families and reduce costs. Although algorithms can be used to optimize product families, they provide point-based solutions and have long computation times.

This thesis presents an approach to cost optimize product families. It combines both a technical and an economical perspective to identify a cost-optimized product family design which fulfills all customer requirements. It solves the so-called joint problem, i.e., it optimizes both the commonality and the component design. The approach consists of five steps: (1) collecting the requirements; (2) modeling and (3) quantifying technical dependencies; (4) modeling costs; and (5) identifying an optimized product family. Step (5) includes two algorithms that a user can choose between: a fast-calculating algorithm that searches in specific areas of the design space (PFD-SSE); and a slower literature-based algorithm that searches the complete design space to find the optimized product family design (PFD-CDS). The developed approach was applied to three use cases: a simple transmission use case illustrating the application of the approach; an electric vehicle use case and a water hose box use case showing real-world applications.

In testing the developed approach, PFD-SSE identified a technically and economically robust solution for the electric vehicle use case and optimized the family of water hose boxes 275 times faster with 13% higher cost compared to existing approaches, represented by the PFD-CDS algorithm. PFD-SSE leads to solutions with higher cost and more robustness, whereas PFD-CDS identifies solutions with lower cost and less robustness.

1 Introduction

1.1 Initial Situation

Krause and Gebhardt (2018) name four trends that affect companies: globalization, new consumer behavior, individualization, and innovation dynamics.

Globalization leads to a fragmentation of value chains. Companies and customers buy international products. Companies can have sales markets all over the world. Different markets have different customers and regulations. Thus, the requirements on 'global' products can be manifold. Furthermore, companies have to face increasing competition by companies across the world. To remain competitive companies need to develop their products fast, cost-efficiently and in high quality (Krause and Gebhardt, 2018).

A *new consumer behavior* can be especially observed in the western countries. After a phase of increasing demand, consumers begin to focus more on quality and sustainability. This goes along with an emphases on *individualized* products (Krause and Gebhardt, 2018). Before the industrialization, craftsmen created products according to the individual needs of their customers. This product development and production process is expensive. With the beginning of the industrialization, companies succeeded by producing large amounts of standardized products at low costs. They decoupled development and production of their products. They developed a product and produced it with no or little changes in a small variety for some years. Henry Ford is one of the most famous representatives of that era. He has been quoted: "any color you want – so long as its black" (Simpson et al., 2006, p. 1). Now industry has changed again. Companies are trying to fulfill individual customer needs at the production volumes of mass production. Pine (1993) calls this phenomenon mass customization. It challenges companies to offer individual products at low costs. Developing countries are at the beginning of the change the western countries have undergone (Krause and Gebhardt, 2018). They are rather focused on consumption and growth. These contradictory customer wishes diversify the product requirements and challenge companies to offer products on a global market.

A fast and inexpensive exchange of information fosters the gain of knowledge. This knowledge finds its way faster into new products (Krause and Gebhardt, 2018). *Innovation dynamics* increase rapidly. As new technology is developed and integrated into products, the complexity of these products rises. Designers from different disciplines need to work together to develop more complex products with smaller lifetimes (Krause and Gebhardt, 2018).

Thus, companies have to develop **complex** products with a **wide variety of customer requirements** while being subject to **high cost pressure**.

Some companies try to enter new markets to increase sales and mitigate the cost pressure. This can lead them into the vicious circle of diversification (Krause and Gebhardt, 2018). The high cost pressure limits the competitiveness of companies because it reduces their profit. They want to use their resources more efficiently. Thus, some companies widen their portfolio to satisfy more customers in new markets. Individual demands and dynamic requirements intensify the process of expansion of the offer. The companies introduce new product variants to meet the requirements of new customers. New product variants often increase the variety on the component side. This increases the complexity in the company. Increasing complexity reduces transparency and vice versa. The companies, e.g., need to maintain an increasing number of part identification numbers which makes it difficult to look for and find existing components. Thus, the reuse of components decreases and development costs rise. The increased complexity drives up costs. Diversification reduces production batch sizes (Wortmann et al.,

1996) and at the same time increases direct and indirect costs (complexity costs) (Ehrlenspiel et al., 2020). The cost per units rises. Rising costs reduce the competitiveness of the company. The benefits from the sale of new variants often do not keep pace with the arising costs of new variants (Jiao et al., 2007). The vicious circle of diversification starts again. Figure 1.1 summarizes this vicious circle.

1.2 Motivation

Product development can break the vicious circle of diversification (see figure 1.1). It needs to reduce the increase of component variety due to the introduction of new product variants. A new product variant should not introduce automatically new component variants – or put differently: increasing external variety should not automatically lead to increasing internal variety. At the same time, products are becoming more complex, i.e., they consist of many different components. If new variants of these components are introduced the internal variety of components rises. The number of part numbers in industry increased drastically (Ehrlenspiel et al., 2020). The main problem of the high internal variety are the high costs (Ponn and Lindemann, 2011). Thus, many companies invest in product family development to offer variety at reasonable cost (Robertson and Ulrich, 1998; Simpson et al., 2014).

The companies need to switch from the design of single products to the simultaneous design of entire product families (Krause and Gebhardt, 2018). They have to master the internal variety while offering a sufficiently wide external variety (Ehrlenspiel et al., 2020). They have to identify potential for standardization from the very beginning of the design process. Standardization, i.e., the re-use of components is one major lever for companies to reduce internal complexity (Simpson et al., 2006) and stay competitive (Robertson and Ulrich, 1998). It can mitigate the impact of high external variety: less activities in design, documentation, test and certification lead to lower variant-related complexity costs, as well as savings in tooling, machinery, storage, logistics, distribution, maintenance and warranty. Increasing volumes enable economies of scale. Thus, it can also help them to build new products faster, cheaper and more robustly (Simpson et al., 2006; Robertson and Ulrich, 1998).

However, standardization comes at a prize. The re-use of components across different product variants with different requirements typically requires an overhead in functionality and, thus, over-dimensioning (Fujita, 2002). Over-dimensioning means that a component is designed for stricter requirements than needed for the product variant: Consider for example a motor to be shared between two vehicles. The vehicles need to accelerate in a certain time and drive at least a certain velocity. If one vehicle has stronger requirements on acceleration and velocity the motor will be over-dimensioned for the other vehicle. This effect diminishes the benefits of standardization. An individual motor

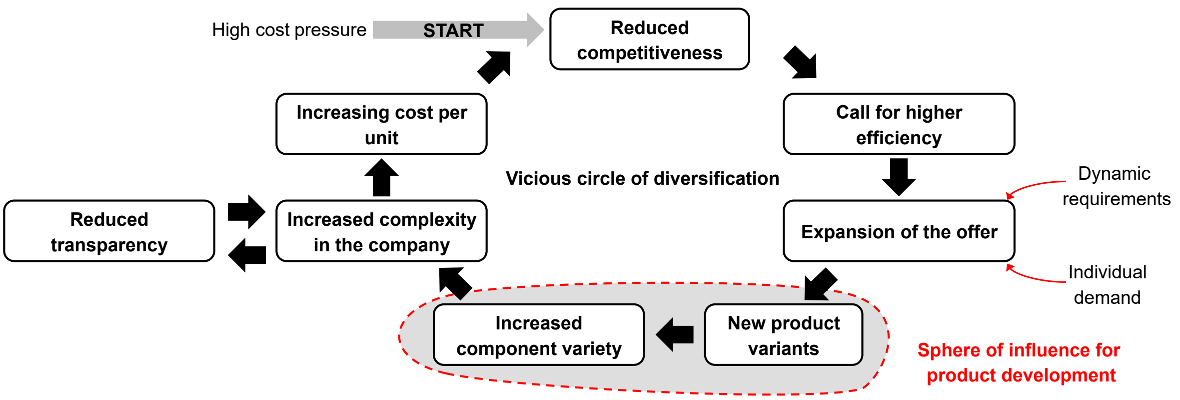


Figure 1.1: Vicious circle of diversification. Adapted from Krause and Gebhardt (2018, p. 7)

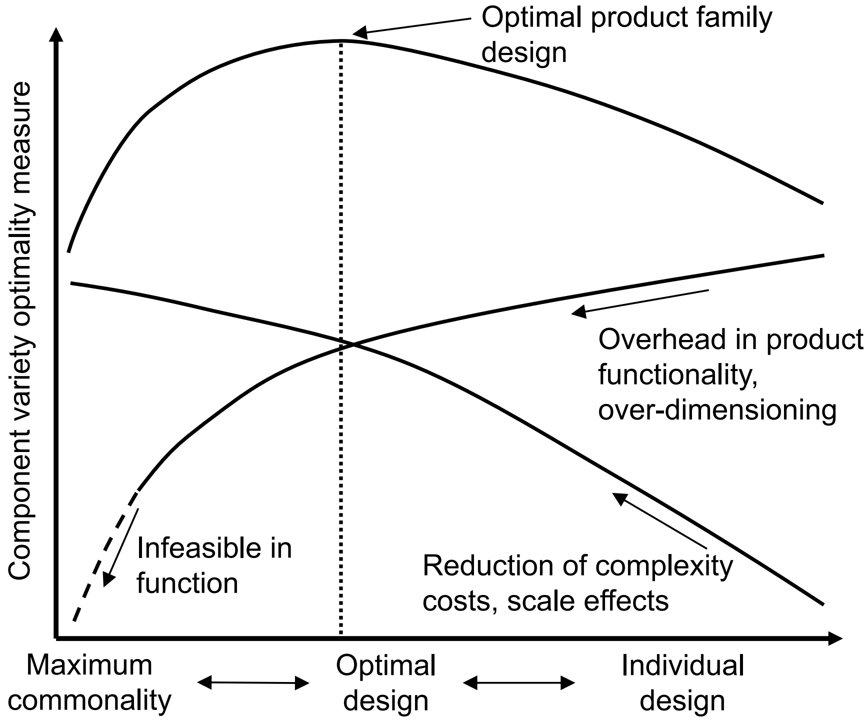


Figure 1.2: Trade-off of standardization. Adapted from Fujita (2002, p. 456)

design could scale the motor down and, e.g., use less material to save costs.

There are two major effects of standardization which are in conflict with each other: reduction of complexity costs and increasing scale effects vs. over-dimensioning of the components. Figure 1.2 shows this trade-off of standardization. The x-axis indicates the level of commonality/standardization. On the left there is maximum commonality; on the right the individual design of each component for one product variant. The y-axis is a component variety optimality measure of the type 'bigger is better'. A possible measure can be cost savings. The descending curve indicates the benefits of standardization, such as reduction of complexity costs or scale effects. The ascending curve indicates the drawbacks of standardization, such as over-dimensioning. The curve on the top accumulates both effects and reveals an optimum between maximum commonality and individual design. Maximum commonality is technically limited at some point because in most cases it is not possible to use only one component variant for the entire product family and fulfill all requirements. As commonality is "intentional - not coincidental" (Simpson et al., 2014, p. 5), product family design is a design task. The goal is to find the optimized degree of standardization of the components of a product family. But designing and optimizing an entire product family instead of one product increases complexity (Simpson et al., 2006). This challenges companies and requires a different mindset: Designers have to think in product families not only single products (Simpson et al., 2014). The high number of possibilities to share component variants across a product family makes it difficult for designer to identify the optimal degree of standardization. The Bell number calculates the increase of component variety (number of component variants) in the product family with increasing number of product variants and components (Baylis et al., 2018).

Equation (1.1) is the recursive definition of the Bell number B_{n_P} (Bell, 1934). It represents the number of possibilities to share variants of *one* component in a product family of n_P product variants. Equation (1.2) calculates the number of possibilities to share variants of n_C components in a product family. It increases exponentially with the number of components n_C.

$$B_{n_P} = \sum_{k=1}^{n_P} \binom{n_P - 1}{k - 1} B_{k-1} \tag{1.1}$$

$$B_0 = 1$$

$$B_{PF} = (B_{n_P})^{n_C} \tag{1.2}$$

Table 1.1 gives an impression how the number of possibilities to share component variants within a product family increases. There are already 877 possibilities to share variants of *one* component within a product family of seven products. If the product family consists of five components this number increases to 519 trillion possible combinations.

This combinatorial problem of assigning component variants to product variants challenges designers. At some point, it is impossible to design and optimize a product family without method and tool support. However, at the same time the high number of possible combinations to share component variants within a product family also challenges optimization algorithms. Although those algorithms can already achieve remarkable results there is room for improvement (Simpson et al., 2014).

Complex products include different disciplines. Thus, it is a multidisciplinary design optimization (MDO) problem. The main motivation for MDO is to include the interactions of the different disciplines to find the best design (Martins and Lambe, 2013). However, in many companies employees do not think holistically but discipline oriented (Albers and Gausemeier, 2012). Models used in companies to describe the system behaviour of complex products are often discipline specific. "Integrating them into a coherent whole requires a special effort" (de Neufville and Scholtes, 2011, p. 107). These models are often dependent on the person who made it and, thus, poorly documented, large and difficult to understand (de Neufville and Scholtes, 2011). They do not support intuitive thinking. There is a need for augmenting intuition with models (de Neufville and Scholtes, 2011). The designer should be able to work with the models to explore the design space and include implicit knowledge.

The future is uncertain (de Neufville and Scholtes, 2011). This requires a flexible and robust product

n_P \ n_C	1	2	3	4	5
1	1	1	1	1	1
2	2	4	8	16	32
3	5	25	125	625	3125
4	15	225	3375	50625	759375
5	52	2704	140608	7311616	380204032
6	203	41209	8365427	1698181681	3.44731E+11
7	877	769129	674526133	5.91559E+11	5.18798E+14

Table 1.1: Number of possible combinations to share n_C component variants in n_P product variants calculated according to the Bell number (Bell, 1934)

(family) design. Classical optimizations are based on one contextual situation. Uncertainty, however, requires optimizations for different contextual situations and projections of the future. An optimized design of one contextual situation can behave badly in another. This is especially true for product families as their design is closely linked to production volumes and economies of scale. de Neufville and Scholtes (2011) call these optimizations "naive". There is a need for robustness and flexibility in design optimization.

However, the optimization algorithm is only one part of the solution. As shown in the vicious circle (see figure 1.1), mastering complexity and providing transparency are also interesting areas for the development of new methods.

1.3 Goal and Structure of the Thesis

The goal of this thesis is to develop an approach to design cost-optimized product families. The cost effects relevant for standardization should be considered. All requirements of the product variants must be fulfilled. The approach should be able to optimize both the assignment scheme (how component variants are shared within the product family) and the the design of the component variants. It should guide users through the process to enhance applicability. It should be able to minimize costs, maintain the external variety and help to master complexity. The following questions structure the overall goal. They are assigned to the chapters in this thesis where they are answered.

Q1	What is the state of the art in product family optimization?	Chapter 2
Q2	What are the challenges for companies in product design?	Chapter 3
Q3	What must a new approach look like to meet academia and industry-related challenges?	Chapter 4 & 6
Q4	How does the developed approach perform in application?	Chapter 7 & 8

Figure 1.3 gives an overview of the structure of the thesis. *1 Introduction* motivates the problem and gives an overview of the thesis. *2 Fundamentals of Product Family Design* includes a literature review and a definition of key words. It positions the topic of the thesis in the field of product family design and compares state of the art approaches to identify the research gap and academia-related challenges. *3 Empirical Studies* gives insights in current challenges of product design. It contains information from a workshop and a interview study. It reveals industry-related challenges. Both academia and industry-related challenges are used to derive *4 Requirements of a New Approach*. *5 Fundamentals of the Developed Approach* incorporates all the background knowledge from existing approaches used in the developed approach. *6 Developed Approach* provides all necessary information for the application of the approach. It formulates the prerequisites and explains the method illustrated by an introductory example. The method consists of five steps including a method to automatically assemble modular models to a system model and two optimization algorithms the user can choose from. *7 Evaluation* uses two industry cases to illustrate the application to two real world problems. *8 Discussion* uses the the evaluation cases from chapter 7 to discuss the fulfillment of the requirements from chapter 4. *9 Summary and Outlook* concludes the thesis by summing it up and provides impulses for further research in the outlook.

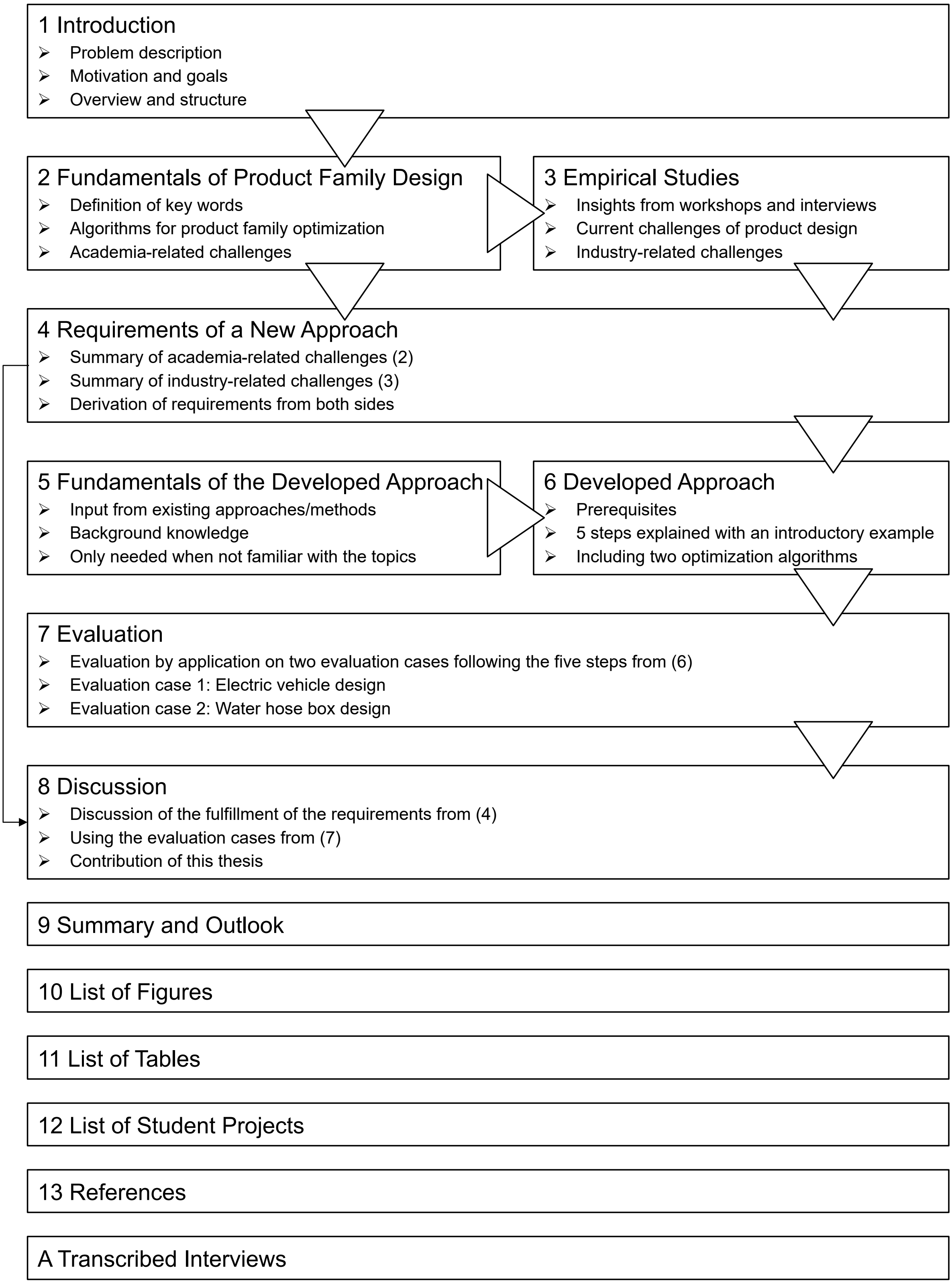


Figure 1.3: Structure of the thesis

2 Fundamentals of Product Family Design

This chapter summarizes and categorizes related literature. It starts with the definition of key words in section 2.1. Section 2.2 gives an overview of the topics of product family design and sets the scope.

2.1 Definition of Key Words

Due to numerous research activities and applications in different companies, different definitions of key words related to product family design exist. Companies can use different terminologies than academia (Krause and Gebhardt, 2018). Simpson et al. (2014) claims a need for standardization of those definitions to increase effectiveness in design. This section defines the key words used in this thesis.

Kotler and Armstrong (2013) consider a (physical) *product* as a part of a market offering to satisfy a need or a want. It is crucial for the success of a product that it fulfills the requirements by customers (Ponn and Lindemann, 2011). Companies can offer a variety of products on the market to satisfy the needs of different customers (Ulrich, 1995). Kirchner (2020) calls this variety of product variants that can be used directly by the customer *external variety*. According to Krause and Gebhardt (2018) (in reference to Rupp (1988)), the *product portfolio*[1] summarizes all products and services that a company offers on the market. Figure 2.1 shows the classification of a product portfolio. It consists of own products (production program[2]), bought-in products and services. A company can organize their production program in product lines. *Internal variety* describes the variety of components, assemblies, products and processes that occur within a company (Kirchner, 2020). This high number of parts that need to be handled makes it difficult for companies to maintain an overview. They may not know which component variants they have already developed. Variant management comprises the handling and influencing of variety (Ponn and Lindemann, 2011). As a possible solution, Simpson et al. (2014, p. v) proposes product family design "to achieve sufficient variety for the marketplace while remaining cost-effective and competitive".

Table 2.1 presents definitions of a product family. Each individual product within a product family can be called a *product variant* (Jiao et al., 2007). A product family is often considered as a set of product variants that must fulfill requirements by certain customer groups. They incorporate similar product functions and a similar product structure. Ulrich (1995), Feldhusen and Grote (2013), Schuh and Riesener (2017) name the combination of functional and physical structure of a product product architecture.

In this thesis, a *product family* consists of product variants with a similar product architecture. The product variants that constitute the product family differ from each other according to the requirements they have to fulfill. Components are defined very differently in different contexts (Krause and Gebhardt, 2018). They can be parts, assemblies or sub-systems depending on the scope. This thesis follows the recommendation of Krause and Gebhardt (2018) and considers a *component* as the smallest considered unit. Components can be shared among a product family. This is called *commonality*. Wazed et al. (2010) give an overview of different definitions of commonality in different contexts with a focus on manufacturing. The *assignment scheme* designates which components are common among the product variants. In this thesis different variants of a component which can be shared among a product family are called *component variants*. There are other interpretations of commonality. Reitan

[1] translated from German: 'Produktprogramm'
[2] translated from German: 'Produktionsprogramm'

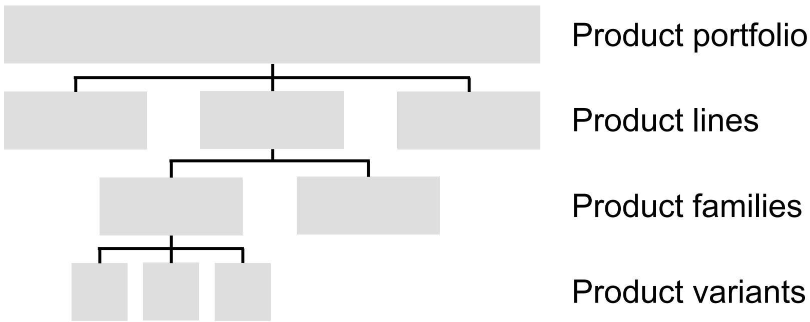


Figure 2.1: Position of a product family within a product portfolio. Figure adapted from Krause and Gebhardt (2018)

Reference	Definition of product family
Meyer and Utterback (1992, p. 3)	"products that share a common platform but with different features and functionalities required by different sets of customers"
Meyer (1997, p.17)	"products that share common technology and address related market applications."
Meyer and Lehnerd (1997)	a set of similar products that are derived from a common platform and yet possess specific features/functionality to meet particular customer requirements
Agard (2004)	represents customer groups. Product families based on similar requirements, functional and technical structure
Fellini et al. (2006)	a set of product variants
Simpson et al. (2014)	a set of products that share a number of common components and functions with each product having its unique specifications to meet demands of certain customers

Table 2.1: Overview of definitions of a product family

et al. (2002) define commonality in a broad sense as the reuse of the product itself or linked activities or processes. Erens and Verhulst (1997) define commonality as components that do not have any variants. This is means that all product share the same component, i.e., there is only one component variant.

Jiao et al. (2007) link this concept of a single component variant to the term platforms in their literature review (Erens and Verhulst (1997) do not use the term 'platform' in the original paper). Table 2.2 gives an overview of different definitions and points of view of product platforms. Most of them rely on the idea of sharing components or other properties, such as functionality or production aspects. The points of view and definitions of a platform vary significantly (Simpson et al., 2006). It can easily lead to misunderstandings: One may regard the platform as the largest possible standard module (Gebhardt et al., 2016), while another thinks of a set of modules which create product variants through configuration (Meyer and Lehnerd, 1997). Simpson et al. (2006) consider defining a platform as one of the most challenging aspects of product family design. Sometimes the terms product family design and platform design are used interchangeably. This thesis considers a *platform* as a set of

Reference	Definition of platform
Meyer and Utterback (1992)	They replace the term product platform by product core to emphasize a broader scope than physical components only. It is the heart of a successful product family, serving as the foundation for a series of closely related products.
McGrath (1995, p. 39)	"a collection of the common elements, especially the underlying core technology, implemented across a range of products"
Meyer (1997)	The foundation of core technology.
Meyer and Lehnerd (1997) according to Jiao et al. (2007)	A modular platform can create product variants through configuration of existing modules
Robertson and Ulrich (1998, p. 20)	"the collection of assets that are shared by a set of products". These assets can include components, processes, knowledge and people.
Simpson et al. (2001)	A scalable product platform can efficiently create variants of same functions by dimensioning.
Fellini et al. (2006)	the set of common parts in a family
Jiao et al. (2007, p. 7)	"All product variants share some common structures and product technologies, which form the platform of the product family."
Gebhardt et al. (2016)	the largest possible standard module in a product family, which is the same for all product variants.
Kirchner (2020)	a set of subsystems and interfaces that form a common structure from which products are derived.

Table 2.2: Overview of definitions of a platform

those components which have only a few variants, i.e., a very high level of commonality. This is closely related to Gebhardt et al. (2016), where the platform is the largest possible standard module in a product family. But to avoid confusion, the term 'platform' is not used in the methodology described here. The methodology in this thesis supports product family design by optimizing commonality. The user can then call these common components a platform or not. Platforms can then build the basis for developing the next generation of products more efficiently (Martin and Ishii, 2002). Product family design is used to utilize commonality (Jiao et al., 2007).

2.2 Overview and Scope of the Thesis

> "Product family design and development has been tackled from various perspectives as well, such as in the areas of business strategy, marketing, manufacturing and production, customer engineering, information technology, and general management" (Jiao et al.,

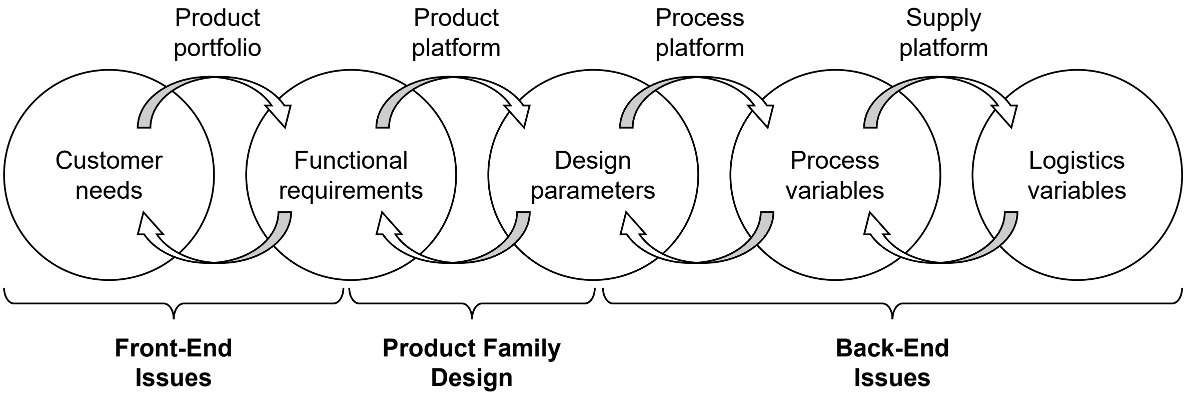


Figure 2.2: Overview of domains of product family design. Figure according to Jiao et al. (2007)

2007, p. 6).

Therefore, this section provides an overview of the fields of product family design along the categories: domain, design direction, design for variety, type of product family, and type of commonality. It sets the scope for the detailed literature review related to the approach presented in this thesis.

Domain

Figure 2.2 gives an overview of the different domains of product family development. *Front-end issues* deal with customer-related topics. According to Jiao et al. (2007), *product family design (PFD)* is the deduction of design parameters from the functional requirements. *Back-end issues* summarizes aspects of manufacturing and logistics.

Approaches for front-end issues provide functional requirements and a product portfolio as a result. They focus on the customers and their needs. Some of them deal with the question which variety needs to be offered on the market to satisfy a certain set of customers. Simpson et al. (2014) calls this product portfolio positioning. Examples are: Kazemzadeh et al. (2008) uses quality function deployment (QFD) and clustering to derive functional requirements for different market segments; Doraszelski and Draganska (2006) describes market segmentation strategies mathematically considering the trade-off between the positive and negative aspects of market segmentation. Kipp and Krause (2009) show how optimization algorithms can be used to define size ranges of customer-relevant product characteristics for a given level over-dimensioning and number of variants. The evolutionary algorithm used in this reference can lead to a higher percentage of satisfied customer requests compared to manually defined size ranges. The quality of manually developed size ranges decreases with increasing complexity of the problem.

Simpson et al. (2014) also names the selection of platform variables and platform configuration a front-end issue. This includes finding optimal platform parameters through algorithms and module identification through data mining techniques, reasoning systems or clustering approaches. Furthermore, approaches for knowledge integration about product families are part of front-end issues. This can include: unified modeling approaches, graph grammar approaches, architecture modeling, set-based models, parametric modeling, functional models (Simpson et al., 2014).

While front-end issues focus on the customer, back-end issues incorporate rather manufacturing related topics. This can include standardization and commonality of manufacturing processes, facilities and technologies. Controlling of manufacturing costs and process is also a topic. Other approaches optimize the supply chain according to profit, delivery time or assembly costs. Postponing the emergence of variety in the production process is another lever in product family design (Feitzinger and Lee, 1997). Therefore, approaches redesign product architectures (Simpson et al., 2014). Cormier et al. (2009) proposes an approach for design product families re-configurably to reduce product family cost by adding flexibility.

Other approaches consider the whole supply chain. Wang and Ning (2007) study the influence of modularization and standardization in the context of the supply chain of product families and derive a model with stakeholders and tasks. Fujita et al. (2013) combine module commonalization with supply chain configuration mathematically. They optimize the product family according to their overall cost by varying the assignment scheme and supply chain related variables, such as production volumes in distributed manufacturing and assembly sites.

Design directions
Product families can be designed along two directions: (1) *bottom-up* and (2) *top-down*. Bottom-up or reactive approaches take existing products and consolidate the the internal variety. In top-down approaches companies proactively design their product family. Products are deduced systematically from strategically defined product families to fulfill a variety of customer needs (Simpson et al., 2014).

Design for variety
Du et al. (2001) introduces two views on product families and variety: (1) *functional variety* and (2) *technical variety*. On the one hand, functional variety is needed to satisfy customers and should be high. This corresponds to the term *external variety* described in section 2.1. On the other hand, product family design should reduce technical variety. Technical variety corresponds to the term *internal variety* and is relevant for manufacturability and costs. "The two types of variety motivate two different design strategies" (Jiao et al., 2007, p. 9): (1) aims to increase external/functional variety to reach more customers and increase revenue, (2) aims to decrease internal/technical variety to decrease costs. In this thesis, product family design should enable a given high external variety with as little costs as possible. This typically requires little internal variety.

Type of product family
Simpson et al. (2006), Jiao et al. (2007) and Simpson et al. (2014) categorize product family design in two groups: (1) *configurable, module-based* product families; (2) *parametric, scale-based* product families.

In a *module-based* product family the configuration of modules (adding, removing or substituting) creates product variants. The combined modules provide then a certain functionality to the customer. This combining becomes easier as soon as the modules can be changed independently and there is a one-to-one mapping of function to module (Ulrich, 1995). Salvador (2007) gives an overview of definitions of modularity citing nearly 50 references. Module-based product family design aims to minimizes interactions between modules, while interactions in a module may be high (Ulrich, 1995). The goal is to provide product variety to customers by combining mostly standardized modules and standardized interfaces. Consequently, the three main tasks in module-based product family design are: (1) modularization and module identification; (2) interface standardization; and (3) architecture embodiment (Jiao et al., 2007). Approaches are often matrix-based (e.g., quality function deployment (QFD)/Modular Function Deployment™(MFD™) (Ericsson and Erixon, 1999); design structure matrix (DSM) (Yu et al., 2003) and rely on clustering techniques. Holtta and Salonen (2003) applies matrix-based approaches on four commercial products and compares them according to the results. Gershenson et al. (2003) provides an overview of modular product design. They usually need measures for modularity.

Scale-based or parametric product families "'stretch' or 'shrink' the platform in one or more dimensions to satisfy a variety of market niches" (Simpson, 2004, p. 5). The design variables are used to scale components to create new products with different performances that fulfill the functional requirements. According to Jiao et al. (2007) and Simpson et al. (2006), scale-based product family design involves two tasks: (1) platform selection and (2) optimizing scalable design variable values.

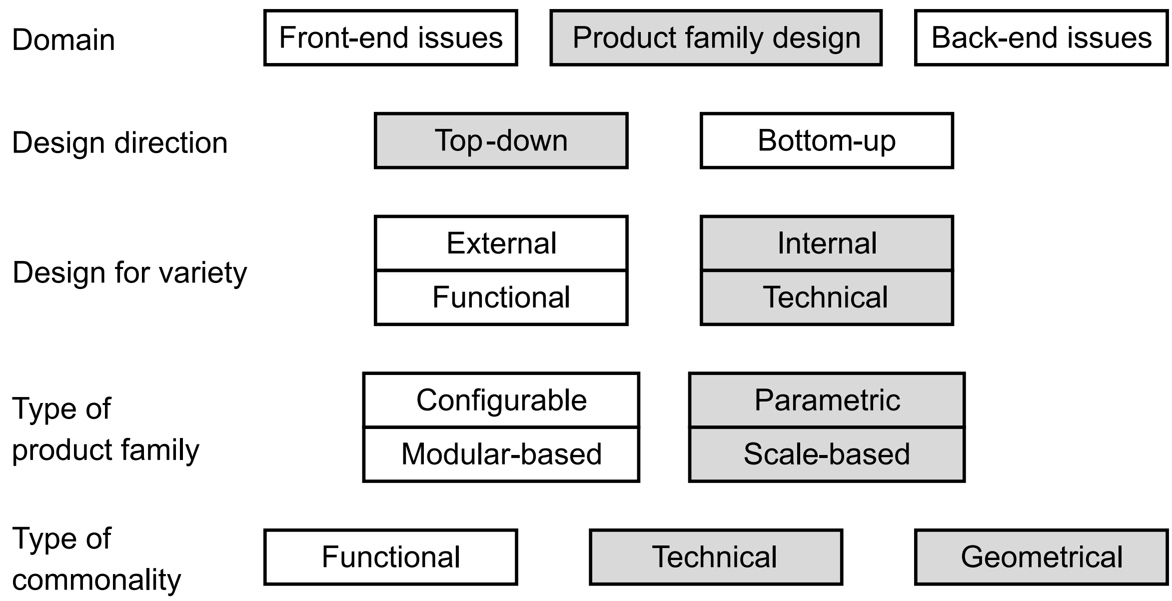


Figure 2.3: Categories of product family design and scope of this thesis (grey).

(1) selects those design variables which should take common values. (2) determines optimized values for shared and distinctive design variables subject to performance and/or economical measures. Due to the parametric problem statement, most researchers use optimization algorithms (Simpson et al., 2006). They optimize design variable values according to objectives subject to constraints.

Type of commonality
There are three types of commonality (corresponding to the domains depicted in Figure 2.2): (1) *functional commonality* (according to the functional requirements, i.e., solution-neutral grouping of customers' needs represented by functional requirements), (2) *technical commonality* (according to technology how the functions are realized, i.e., technical feasibility), and (3) *physical commonality* (according to manufacturability and physical interactions) (Jiao et al., 2007).

Figure 2.3 summarizes the categorizes of product family design and describes the scope of this thesis: a top-down, scale-based product family design approach, which decreases the internal variety by optimizing the technical and geometrical commonality .

2.3 Algorithms for Scale-based Product Family Design

To optimize scale-based product families, approaches typically rely on algorithms (Simpson, 2004). This section classifies those algorithms and gives an overview of the state of the art.

2.3.1 Overview and Scope

Figure 2.4 classifies existing optimization algorithms according to three dimensions: commonality, platform selection and variant design.

Commonality
Commonality is divided into *restricted* and *generalized* commonality. Restricted commonality allows

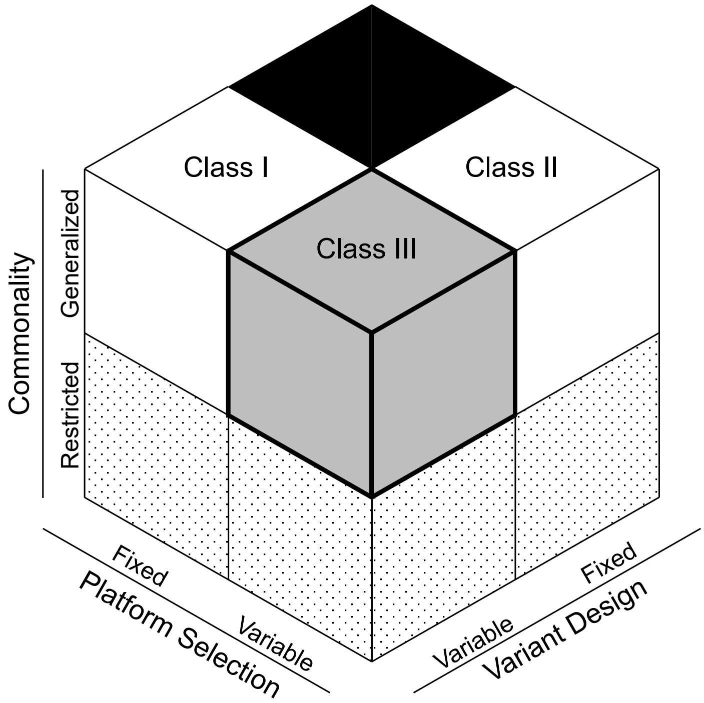


Figure 2.4: Classification of optimization algorithms for product family design and scope of this thesis (grey). Figure according to Simpson et al. (2014).

	Product A	Product B	Product C	Product D	Product E
Component 1	1.1	1.1	1.2	1.2	1.2
Component 2	2.1	2.2	2.3	2.4	2.5
Component 3	3.1	3.1	3.1	3.1	3.1

Figure 2.5: Generalized commonality for component 1; restricted commonality for component 2 and component 3

a component only to be common in all products (see component 3 in Figure 2.5) or none (see component 2 in Figure 2.5). Approaches, such as Simpson (2004), Liu et al. (2011), and Wei et al. (2019), use restricted commonality to reduce computational effort. However, de Weck (2006) shows that restricted commonality leads to worse results in costs, revenue and performance compared to a product family which allows also a partial commonality. This means that a product family consists of several component variants which can be shared among the members of the product family. Product A and B share component 1.1 while product C, D, E share component 1.2. (see component 1 in Figure 2.5). This is called *generalized commonality* and the focus of this thesis.

Platform selection and **variant design**
The other two dimensions are platform selection and variant design. Platform selection describes the task of determining the assignment scheme. Variant design optimizes the design of the components of a product family. Both dimensions distinguish between *fixed* and *variable*.

By combining those three categories Fujita (2002) identifies three classes of optimization algorithms: class I-III. Class I algorithms optimize variants for a predefined assignment scheme (see Messac et al. (2002) and Farrell and Simpson (2003)). Those algorithm solve a continuous problem, as the design variables of the optimization problem are the attributes of the components. Class II algorithms optimize assignment schemes for a predefined set of components, i.e., the design of the components is known a priori (Simpson et al., 2006). Those algorithms solve a combinatorial optimization problem which often has a integer structure. It has to determine whether a component is shared or not within the product family or within a subset. Several studies have been conducted on class I and II problems. Simpson et al. (2014) provide an structured overview. Class III algorithms do not require a priori knowledge of the assignment scheme nor the component design. They optimize both the integer assignment scheme and continuous component design simultaneously. Thus, they solve a mixed integer, non linear problem (MINLP). This makes the development and application of those algorithms challenging (Simpson et al., 2006). Simpson et al. (2014, p. 273) call this problem "joint product family platform selection and design problem" or briefly: joint problem.

This thesis will focus on joint problems (variable assignment scheme and component design).

2.3.2 State of the Art Algorithms

According to Simpson et al. (2014), studies move towards handling joint problems. Algorithms for joint problems need to optimize (1) the assignment scheme and (2) the values for the attributes of the components. (1) incorporates identifying the component variants which should be shared and determining how they should be shared among the product family.

Equation 2.1 shows the typical generic structure of problem statements for product family optimization according to Simpson et al. (2006, p. 23).

$$\begin{aligned} &\mathit{Find} && \mathbf{x} \\ &\mathit{Minimizing} && \mathbf{f}(\mathbf{x}) \\ &\mathit{Subject\ to} && \mathbf{g}(\mathbf{x}) \leq 0 \\ & && \mathbf{h}(\mathbf{x}) = 0 \\ & && \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u \end{aligned} \tag{2.1}$$

With:

- **x**: the decision vector. It describes product related characteristics of the product family, such as the design variables of the components. It can also include the assignment scheme.
- **f(x)**: the performance or objective function. It assesses the product family quantitatively. Can consist of one or more objectives, such as: product family cost or profit, a commonality measure or the market differentiation.
- **g(x)** and **h(x)**: the inequality and equality constraint functions. They can apply requirements on products or enforce a specific assignment scheme.
- $\mathbf{x}^{l,u}$ the lower and upper bounds of the considered design space.

Figures 2.6 and 2.7 provide an overview of relevant optimization algorithms. This list is a result of a systematic literature review using the Scopus database and a cross-reference search.

Handling commonality

Eleven approaches of the 26 approaches listed in figures 2.6 and 2.7 restrict their commonality to reduce combinatorics and thus the computational effort of the optimization problem. Three approaches solve a problem of class I, i.e., they optimize only the design variables of the components and not the assignment scheme. One approach focuses on the back-end issues of product family design.

Number	Authors	Title	Year	Class	Domain	Robust	Decision variables	# objectives	Objective function	Handling multi objectives	Commonality	Levels	Algorithms
1	Fujita, K.; Yoshida, H.	Product Variety Optimization Simultaneously designing Module Combination and Module Attributes	2004	III	PFD	0	DV, Com	1	Profit	–	G	3	GA, Branch and bound, sequential quadratic prog.
2	Simpson, T.W.; D'Souza, B.S.	Assessing Variables Levels of Platform Commonality within a Product Family using a Multiobjective Genetic Algorithm	2004	III	PFD	0	DV, Com	2	Perf, Com	Pareto	R	1	GA
3	Fellini, R.; Kokkolaras, M.; Papalambros, P.Y.	Quantitative platform selection in optimal design of product families, with application to automotive engine design	2006	III	PFD	0	DV, Com	2	Perf, Com	Thresholds	G	1	Clustering, gradient-based optimization
4	Dai, Z. ; Scott, Michael J.	Effective Product Family Design Using Preference Aggregation	2006	III	PFD	0	DV, Com	2	Perf	Preference Aggregation	R	3	Preference Aggregation; min-max; optmization
5	Khajavirad, A.; Michalek, J.J.; Simpson, T.W.	A decomposed genetic algorithm for solving the joint product family optimization problem	2007	III	PFD	0	DV, Com	2	Perf, Com	Pareto	G	2	GA (NSGA-II) (2)
6	Liu, Z.; Wong, Y.S.; Lee, K.S.	Towards effective multi-platforming design of product family using genetic algorithm	2007	III	PFD	0	DV, Com	2	Perf, Com	Weighting	G	1	GA
7	Chen, C.; Wang, L.	Multiple-platform based product family design for mass customization using a modified genetic algorithm	2008	I	PFD	0	DV	2	Perf	Pareto	G	1	GA
8	Chen, C.; Wang, L.	Modified genetic algorithm for product family optimization with platform specified by information theoretical approach	2008	I	PFD	0	DV	3	Perf	Pareto	G	1	GA
9	Li, Z.; Feng, Y.; Tan, J.; Wei, Z.	A methodology to support product platform optimization using multi-objective evolutionary algorithm	2008	III	PFD	0	DV, Com	2	Perf, Costs	Pareto	R	2	Sensitivity analysis, GA
10	Song, H.; Zhang, Y.; Song, Y.; Wang, Z.; Zhen, L.	Product platform planning: An approach using genetic algorithm	2008	III	PFD	0	DV, Com	1	Com	–	G	1	GA
11	Khajavirad, A.; Michalek, J.J.; Simpson, T.W.	An efficient decomposed multiobjective genetic algorithm for solving the joint product platform selection and product family design problem with generalized commonality	2009	III	PFD	0	DV, Com	2	Perf, Com	Pareto, Weighting	G	1	GA
12	Li, L.; Huang, G.Q.	Multiobjective evolutionary optimisation for adaptive product family design	2009	III	PFD	0	DV, Com	3	Perf, Com	Pareto, Weighting	G	1	GA
13	Wei, W.; Feng, Y.; Tan, J.; Li, Z.	Product platform two-stage quality optimization design based on multiobjective genetic algorithm	2009	III	PFD	0	DV, Com	2	Perf	Pareto	R	2	Sensitivity analysis, GA

PFD: product family design

DV: design variables of components
Perf: performance of product family

R: restricted; G: generalized
Com: commonality

GA: genetic algorithm
PSO: particle swarm optimization

Figure 2.6: Overview of related approaches (Part 1)

Number	Authors	Title	Year	Class	Domain	Robust	Decision variables	# objectives	Objective function	Handling multi objectives	Commonality	Levels	Algorithms
14	Liu, Z.; Wong, Y.S.; Lee, K.S	Modified GQ-based optimizer for multi-objective product family design	2009	III	PFD	0	DV, Com	2	Perf, Com	Pareto, Weighting	G	1	GA
15	Gao, F.; Xiao, G.; Simpson, T. W.	Module-scale-based product platform planning	2009	III	PFD	0	DV, Com	many	Perf	Weighting	R	2	Choose platform DVs; Optimization
16	Chowdhury, S.; Messac, A.; Khire, R.	Comprehensive Product Platform Planning(CP3) Framework	2011	III	PFD	1	DV, Com	2	Perf, Costs	Weighting	G	1	PSO
17	Liu, Z.; Wong, Y.S.; Lee, K.S.	A manufacturing-oriented approach for multi-platforming product family design with modified genetic algorithm	2011	III	Back end	0	DV, Com	2	Perf, Com	Weighting	G	1	GA
18	Wang, W.	Scalable platform design optimization using hybrid co-evolutionary algorithms	2011	III	PFD	0	DV, Com	2	Perf	Pareto	G	2	GA (NSGA-II), PSO
19	Chowdhurry, S.; Messac, A.; Khire, R.	Investigating the commonality attributes for scaling product families using comprehensive product platform planning (CP3)	2013	III	PFD	0	DV, Com	2	Perf, Com	Weighting	G	1	PSO
20	Kristianto, Y.; Helo, P.; Jiao, R.J.	Mass customization design of engineer-to-order products using Benders' decomposition and bi-level stochastic programming	2013	III	PFD	1	Product & process configuration, DV	1	Costs	–	R	2	Shortest Path Problems with Resource Constraints; Benders' decomposition
21	Moon, S. K.; Park, K. J.; Simpson, T. W.	Platform design variable identification for a product family using multi-objective particle swarm optimization	2014	III	PFD	0	DV, Com	7	Perf	Weighting	R	2	ConVio, PSO
22	Eichstetter, M.; Müller, S.; Zimmermann, M.	Product Family Design With Solution Spaces	2015	III	PFD	1	DV, Com	1	Com	–	G	2	Solution Spaces
23	Wei, W.; Ji, J.; Wuest, T.; Tao, F.	Product Family Flexible Design Method Based on Dynamic Requirements Uncertainty Analysis	2017	III	Front end/PFD	1	DV, Com	2	Perf, Com	Pareto	R	2	Sensitivity analysis, NICA
24	Wei, W.; Pingyuan, W.; Zhenyu, T.; Jun, J.; Jin, C.	General Motor Family Flexibility Design Based on NICAII	2017	III	Front end/PFD	1	DV, Com	2	Perf, Com	Pareto	R	2	Sensitivity analysis, NICA
25	Wei, W.; Tian, Z.; Peng, C.; Liu, A.; Zhang, Z.	Product family flexibility design method based on hybrid adaptive ant colony algorithm	2019	III	PFD	0	DV, Com	2	Perf	Pareto, Fuzzy Optimization	R	2	Sensitivity analysis, GA
26	Li, S.; Liu, Z.; Wang, L.; Chen, J.; Ni, M.; Zhang, L.	Full Constraint Parallel Optimization Algorithm for Structural Optimization of Product Family	2019	I	PFD	0	DV	1	Weight	–	R	2	Gradient-based optimization

PFD: product family design
DV: design variables of components
Perf: performance of product family
R: restricted; G: generalized
Com: commonality
GA: genetic algorithm
PSO: particle swarm optimization

Figure 2.7: Overview of related approaches (Part 2)

Khajavirad et al. (2009) map the components to the products in a two dimensional matrix using an index to identify the component variant used in a specific product variant. Each component is described by one design variable. Li and Huang (2009) use generic bills-of-materials (GBOMs) and commonality control genes (CCGs) to describe how products are composed of components and to control the commonality within those components. Fujita and Yoshida (2004) use integer values to indicate whether a component variant is shared (or similar) between two product variants. Chowdhury et al. (2011) build a similar assignment scheme using binary variables in a quadratic matrix of the size: (number of products x number of variables)^2. They do not distinguish between design variables and components within the assignment scheme. Thus, each component can only be represented by one design variable. Simpson (2004) restrict commonality. Thus, "1" indicates a standard component for all product variants, while "0" indicates a individual component. This reduces the number of combination dramatically, but leads to worse results than generalized commonality (de Weck, 2006). Generalized commonality considers all possible assignment schemes.

Another challenge with respect to commonality are components with more than one design variable. To achieve commonality of different component variants, the algorithm has to ensure that all design variable values of the component variants are the same. This also increases complexity of the optimization problem. Therefore, some approaches restrict their components to one design variable only (Eichstetter et al. (2015), Chowdhury et al. (2011)).

Heuristics

20 out of the 26 approaches named above rely on evolutionary heuristics. In 15 approaches a genetic algorithm (GA) optimizes the assignment scheme. Darwin's evolutionary theory inspired GAs. They interpret the design as a string of attributes which can be altered by mutation or recombination (Holland, 1992). The binary structure of the assignment scheme favors the use of GAs. Six approaches also use the GA to optimize the design variables. Two algorithms use a multilevel approach where the GA optimizes the assignment scheme and a subsequent algorithm optimizes the design of the specific assignment scheme (sequential quadratic programming (SQP) in Fujita and Yoshida (2004); particle swarm optimization (PSO) in Wang (2011)). Fellini et al. (2006) relaxes the binary commonality problem to solve it with a gradient-based algorithm. Eichstetter et al. (2015) uses the concept of solution spaces to maximize commonality.

Objectives

Product family design usually incorporates various objectives, such as: modularity, commonality, variety, cost, profit, or performance (Jiao et al., 2007). Scott et al. (2006) state that there is no consensus in the objective functions used. They claim at least two conflicting goals: performance and cost.

Eleven approaches try to solve the conflict of goals between performance of the products and commonality of the products. Performance is always linked with the functional requirements on the products, such as mass or efficiency. The underlying idea of those approaches is that sharing components will reduce the performance of products and thus a conflict of goals needs to be solved.

13 approaches use assumptions on commonality, such as maximizing the commonality (see Fellini et al. (2006), Eichstetter et al. (2015)), or use commonality indices (see Chowdhury et al. (2011), Khajavirad et al. (2009), Liu et al. (2011)). Simpson et al. (2014), Jiao et al. (2007) and Wazed et al. (2010) give an overview of commonality metrics which are used in different approaches. There are many different metrics. Jiao et al. (2007) names 14 references for different commonality metrics. Commonality indices/metrics have in common that they break down the variety of the product family to a scalar value to compare different solutions for the product family with each other. Simpson et al. (2014) propose six commonality metrics. They can be weighted with: mass (relative mass of a component in relation to the mass of the product family); cost (relative production cost per unit); investment (investment required for a component); production volume (relative production volume for

each variant); production volume per investment. Each commonality index has different prerequisites and limitation which influences the result of the optimization. Considering more factors increases the vulnerability of those indices as they become more dependent on further information such as estimation of costs or production volumes (Simpson et al., 2014).

Approaches which rather deal with front-/back-end issues can have very different objectives, such as minimizing production time (Barathwaj et al., 2015).

Approaches with more than one objective (multi-objective optimization) often use Pareto diagrams to display non-dominated solutions or weighting (meta) functions to convert the multi objective problem into a single objective one. The latter usually need to incorporate beneficial and prejudicial effects of commonality.

Overall cost or profit of a product family can be used as a single objective for the optimization. Fujita and Yoshida (2004) model the profit of the product family by considering revenue and cost of the product family. The revenue is dependent on the performance of the products. The cost model consists of: design and development cost, facility cost and production cost. They are dependent on the design variable values, the assignment scheme and the production volumes. In PFD it is important "to what extent the economy of scale can be realized within the existing manufacturing capabilities" (Jiao et al., 2007, p. 16). Thus, the benefits of product family design are realized during production (Jiao et al., 2007).

Some approaches solve a conflict of goals between costs and performance (see Li et al. (2008)). de Weck et al. (2003) proposes a model including profit and cost to design multiple product platforms according to optimized market segmentation strategies. However, P. Tarasewich and S. K. Nair (2001) claim that including the consumer into design process is limited to specific products and to the complexity of a design that can realistically be evaluated. While costs are mostly determined by design variable values, performance measures (or quantities of interest) influence the profit.

Robustness

Five references listed above mention robustness within their optimization approaches. Chowdhury et al. (2011) considers the approach as robust from an application perspective. They use a particle swarm optimization (PSO). PSO performs better for mixed integer non linear problems (MINLP) than gradient-based algorithms. They do not require much adjustment and can be applied straight forward (Chowdhury et al., 2011). Thus, they can be applied easily to different PFD problems. But the approach does not support robust design for a product family.

Kristianto et al. (2013) use Bender's decomposition to solve the back-end optimization problem. It groups the design variables in platform and non platform variables which leads to restricted commonality. Robustness is included by considering different demand scenarios.

Wei et al. (2017a) and Wei et al. (2017b) also apply restricted commonality to divide the design variables into platform and variable design variables. The variable design variables can be used to react flexible to changing requirements. The requirements are modeled with uncertainty. By coupling the uncertainty of requirements with the design variables they introduce a sensitivity measure. This describes how strongly design variables are influenced by changing requirements. This sensitivity should be minimized for the platform.

Simpson et al. (2001) use robust design principles to separate attributes according to their role in the design problem. Then the scaling variables are optimized such that they bring the mean on target and minimize the variation of the performance goals of the system. They use standard deviation to derive designs which are close to the desired goals. They do not develop robust product families per se. Furthermore, the commonality is restricted and pre-defined (class I).

Han et al. (2020) provide a literature review of product platform under uncertainty. They state

Uncertainty-Oriented Product Platform (UOPP) as a third dimension additionally to module and scale-based product families/platforms with the goal to dynamically respond to uncertain factors at low cost. They categorize the approaches into three categories of uncertainty: requirements & technologies; markets (demand & price); requirements & regulations (environment). They state UOPP as a solution for "enterprises to meet uncertain markets, customer requirements, technologies, policies, and regulations" (Han et al., 2020, p. 01).

Schuh et al. (2009) meet uncertainty in requirements and technology by designing an adaptable product platform, which can be changed along changing requirements. Suh et al. (2007) propose a flexible component design for robust product platforms. They use scenarios to model the influence of uncertainty in demand and requirements.

Kumar et al. (2009) present an approach for a market driven product family optimization. They optimize the product family according to the overall profit, including uncertainty on the market side by adding error terms. In a first step, they derive a optimized assignment scheme according to market shares. In a second step, they optimize the design variables according to this assignment scheme. Thus it is considered as a class I approach.

Eichstetter et al. (2015) use solution spaces instead of point-based solutions. This allows for more robustness and flexibility with respect to changing requirements during design. The approach maximizes commonality.

2.4 Challenges and Research Gap

Class III problems are mixed integer non linear problems (MINLP). This challenges optimization algorithms. Some approaches divide the optimization problem in two levels: (1) optimizing the assignment scheme (integer problem); (2) optimizing the component design (non linear problem). Some of them solve those optimizations sequentially, others are coupled. The first do not solve the class III problem, as the design of the components does not influence the assignment scheme. The latter lead to high computational effort. Same applies for monolithic optimization algorithms with one level. Most of them use genetic algorithms to handle the MINLP problem statement, as they are seen to be most suitable for MINLP problems (Simpson et al., 2001).

Another aspect which makes product family design computationally challenging is the increasing number of combinations a component variants can be shared among the product family. For that reason some approaches restrict their commonality to the all-or-nothing commonality. When allowing for a generalized commonality, the possibilities to share component variants within a product family increase drastically with the number of components and the number of product variants. In conclusion:

> "[...] there is a need for an approach capable of solving the joint problem [class III] using generalized commonality (box 6) for practical product family applications with a **reasonable computational cost**." (Simpson et al., 2014, p. 274)

All considered approaches, except Eichstetter et al. (2015), result in a point-based solution. In technical problems, optimized solutions are typically at the limit of constraints. If requirements are uncertain or change, those point-based solutions can become bad solutions. They can violate the updated requirements. Thus, there is a need in an approach, which allows for **robust product family design**. Robust in this context means that the product family still satisfies all requirements when component attributes change a little.

Most approaches use commonality metrics to quantify the influences of commonality. The high number of different metrics, as can be seen in Simpson et al. (2014) or Jiao et al. (2007), show that there is little consensus in which metric to apply. They often incorporate hidden assumptions, such as reducing costs by commonality is independent of the production volume of each product variant (Chowdhury et al., 2011) or that maximum commonality is optimal (Eichstetter et al., 2015). Some approaches try to model profit or cost directly (Fujita and Yoshida, 2004). However, especially the modelling of the profit requires many assumptions on customer behaviour and market predictions. Modelling costs can also be a difficult task as the do not only differ by product but also by company. To conclude: There is still a need in finding an **adequate representation of the effects of commonality** including (uncertain) production volumes.

3 Empirical Studies

This chapter provides experiences and findings from a workshop and interviews with experts from industry. This information lays the foundation for the developed approach from an industry perspective.

3.1 Industry Workshop

A workshop was conducted in February, 2020 at a company providing solutions for heating, cooling and ventilation with around 13 000 employees. Five experts participated from the following fields: systems design and integration; simulation; component design; tools and methods. The goal was to discuss (1) current challenges the company is facing and (2) research ideas.
The experts rated the order of relevance of challenges in the prescribed fields as follows:

1. Variant management
2. Requirements
3. Decisions
4. Complexity

The participants considered requirements as a relevant topic, but they also mention that their is a clear strategy being implemented to enable requirement-driven development. Technical complexity comes into play when considering more products. Thus, it is linked to variant management.

Variant management
Legal compliance and individualization of customers' needs are the main variant drivers. This often leads to a two-digit number of variants per product. Lacking transparency in design induces variants because designers may not know whether existing variants already fulfill the new requirements. The experts seek guidance for the designers to enable design for commonality. Standardization of one component can induce high variety of other components. The influence of standardization on the overall system is unclear due to complexity. Although overall cost is one of the main quantities of interest, knowledge of costs, especially complexity costs, is limited. Variants are often introduced out of necessity due to changing regulations, which have to be met. This leads to high warehousing costs and a low efficiency in production. As standardization often comes along with over-engineering of components, the experts are interested in finding an optimum within: standardization - costs - customer satisfaction.

Decisions
Trade-offs are inherent and occur in every decision. Main decision criteria is the overall cost of a product. But as they are often not known, decisions are made according to experience and gut feeling. Organizational hierarchy dominates the decision-making process. The experts demand quantitative decision support to increase transparency and traceability. The number of simulations increase.

3.2 Interview Study: Current Challenges of Product Design

Building on the experience gained from the workshop (see section 3.1), a systematic interview study was conducted as a student project (Stulpe, 2020). The goal was to get insights from a broader range

	Company	Job	Experience
A	Automotive	Strategic purchasing Goal and requirement engineer	> 1 year > 3 years
B	Online software applications	Product manager	> 3 years
C	Transportation	Project engineer purchase	n/a
D	Manufacturer industrial trucks	Product manager marketing	< 1 year
E	Agricultural machinery	Project coordinator	> 20 years
F	Mechanical engineering	Head of development	> 4 years
G	Technology and electrical engineering	Cost manager purchase	n/a
H	Machine manufacturer	Intern consultant	> 2 years
I	High-end electronics	Project leader	n/a
J	Products for metal cutting	Product manager	> 3 years
K	Mechanical engineering of consumer goods	Simulation engineer	n/a
L	Medical engineering	Head of development Design engineer	> 6 years > 8 years
M	Vehicle manufacturer	Product manager	> 3 years
N	Aviation	Innovation manager	> 1 year
O	Power tools	Technical project leader	> 15 years

Figure 3.1: Overview of the interviewees, their companies and their job. Figure according to Stulpe (2020).

of companies. Those insights guide the development of the approach in this thesis.

3.2.1 Research Design of Interview Study

The interview study conducted here aims at exploration and understanding of the current challenges. Thus, it is a qualitative study - in contrast to quantitative statistical studies. It is not a representative survey because the interviewees were chosen ad-hoc and the sample number is small. Misoch (2019) recommends inexperienced interviewers to use a questionnaire. It fulfills the following functions: (1) Framing of the topics and thus better comparability of the results. (2) Completeness: List of all relevant topics to be discussed. (3) Structuring of the entire communication (Misoch, 2019). The interview was conducted semi-structured to allow for freedom to discuss relevant topics of the experts on the one hand, and on the other hand not to forget about important topics during the interview.
37 experts for product development were contacted, 22 replied, 18 confirmed, 17 interviews were conducted. Those 17 experts belonged to 15 companies with a range from 800 to 380,000 employees. Figure 3.1 gives an overview over the participants and their companies.

Figure 3.2 gives an overview of the topics discussed during the interviews. After a short introduction the participants order the five topics: requirement management; variant management; decision-making; complexity; and robust design according to the relevance in their company. After a discussion on those topics, the interviewer starts the second part of the study by presenting the research ideas. These research ideas are then discussed and evaluated.

The interviews took between 48 and 90 minutes. Due to Covid-19, all interviews were conducted

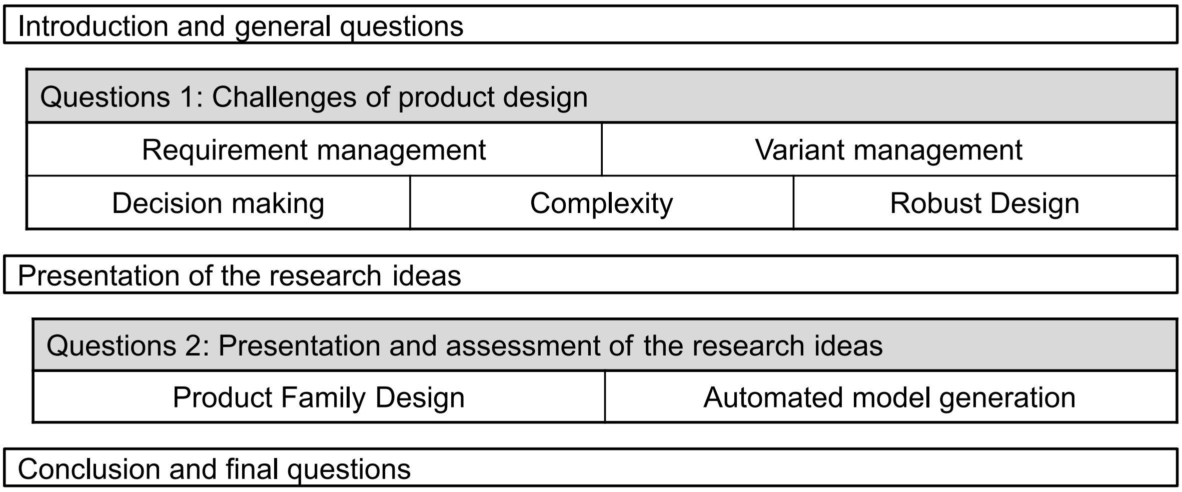


Figure 3.2: Topics of the interview study. Figure according to Stulpe (2020).

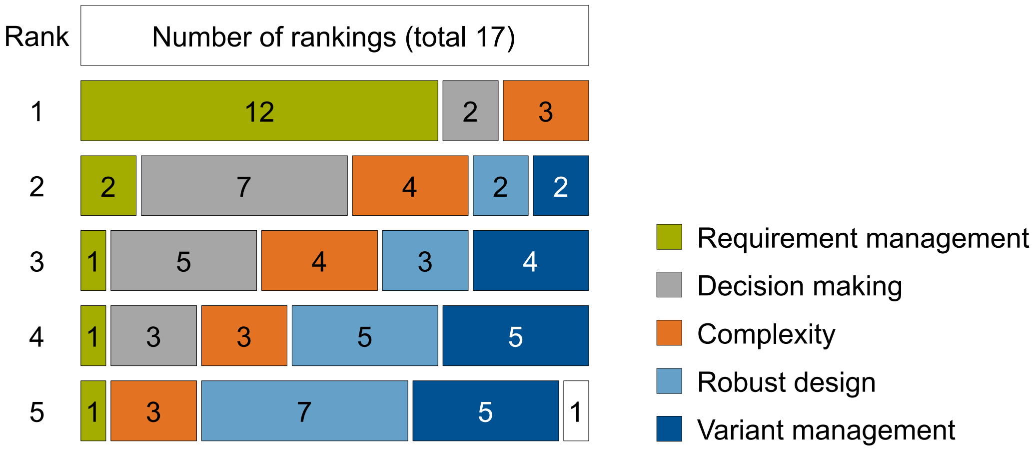


Figure 3.3: Prioritization of current challenges in product development according to the interviewees' rating. Rank 1: highest priority - rank 5: lowest priority. Number indicates number of rankings. Color indicates challenge. Figure according to Stulpe (2020).

digitally as video conferences. They were audio recorded and transcribed for better comparability and traceability. The interviews were conducted in German.

Qualitative content analysis was used to analyze all transcribed interviews (Gläser and Laudel, 2010). Relevant information was extracted systematically and categorized based on the topics of the interview study.

Due to non-disclosure agreements, it is not possible to provide company names in the following.

3.2.2 Challenges of Product Development

Figure 3.3 visualizes how the interviewees prioritized the given challenges. This scheme represents the points of view of the interviewees. It is not statistically representative, but can serve as an overview of current challenges. 12 out of 17 interviewees rate requirement management as the most relevant challenge. Nine out of 17 order decision-making on rank one or two according to its relevance.

“Complexity overarches all topics” [1] as one of the participants said. This can be also seen in the distribution of the answers. Some of the participants struggled to rate complexity independently from the other challenges and perceive it as a possible cause for other challenges. Variant management and robust design was rated differently according to the context of the company. One company for instance develops high-tech equipment for the film industry. There variant management does not play a role at all because of the low sales volumes and customers willing to pay high prices for high performing solutions. Whereas in other companies with more price sensitive markets the importance increases.
The interviewees could also add challenges. This in an excerpt of those: risk management (especially in automotive or medical engineering); technology and technology leaps (e.g., battery development); uncertainty and dynamics (unknown or changing information, especially in the context of decision-making); knowledge management (storing knowledge and information of different people in a re-usable way).
Insights from the different challenges are explained in more detail in the following.

Requirement management

> “...because people do not necessarily mean the same thing when they say the same.”[2]

Late changes in requirements lead to high losses of money. At the same time companies struggle to define the requirements completely in an early phase. They use different sources to elicitate requirements: competitors’ products; law and regulations; predecessors; (market) standards; anticipation of market needs; user tests. Designers and engineers can have implicit knowledge, which lacks to be externalized. Thus, they sometimes may use different wordings and facing difficulties understanding each other. Seven interviewees were asked specifically about conflicts of goals. Four of them mentioned that conflict of goals appear sometimes; three said that they appear every time when working with requirements. One of them mentioned that a project had to be restarted completely from the beginning because a conflict of goals could not be resolved. Customers can also directly influence requirements or decisions due to conflicting requirements, especially when they buy many products. One interviewee names a missing big picture of the product and its requirements as a challenge. The divisions rather work on their own with their scope in mind. Another interviewee claims a theory-guided approach instead of intuition.
Four interviewees said that they could derive requirements from a predecessor. Three companies have the goal to quantify 100 percent of their requirements. At the moment they can quantify approximately 50; 60-70; 75-90 percent respectively. Simulations are used to evaluate designs in an early phase or to optimize designs before production.

Decision-making

> “There are different approaches, and none of them is extremely convincing.”[3]

Typical questions during decision-making are: Which requirements should be prioritized? How will the technology develop? What should the product portfolio look like? How can we develop most profitably?
Figure 3.4 illustrates the methods which are used in the companies to make decisions and their bases. Decision-making in general requires some kind of subjective preferences. But some interviewees complained about the subjective basis on which the decisions are made. Three participants named gut feeling as the basis for decisions; four the term expert knowledge. One persons said that hierarchy

[1] translated from German: "Komplexität hängt über allen Themen drüber" (INr. 0, Z45f)
[2] translated from German: "Weil Menschen nicht unbedingt das Gleiche meinen, wenn sie das Gleiche sagen." (INr. 15, Z.78)
[3] translated from German: "Da gibt es verschiedene Ansätze, und kein[er](d.V.) davon ist extrem überzeugend." (INr. 2, Z.297f)

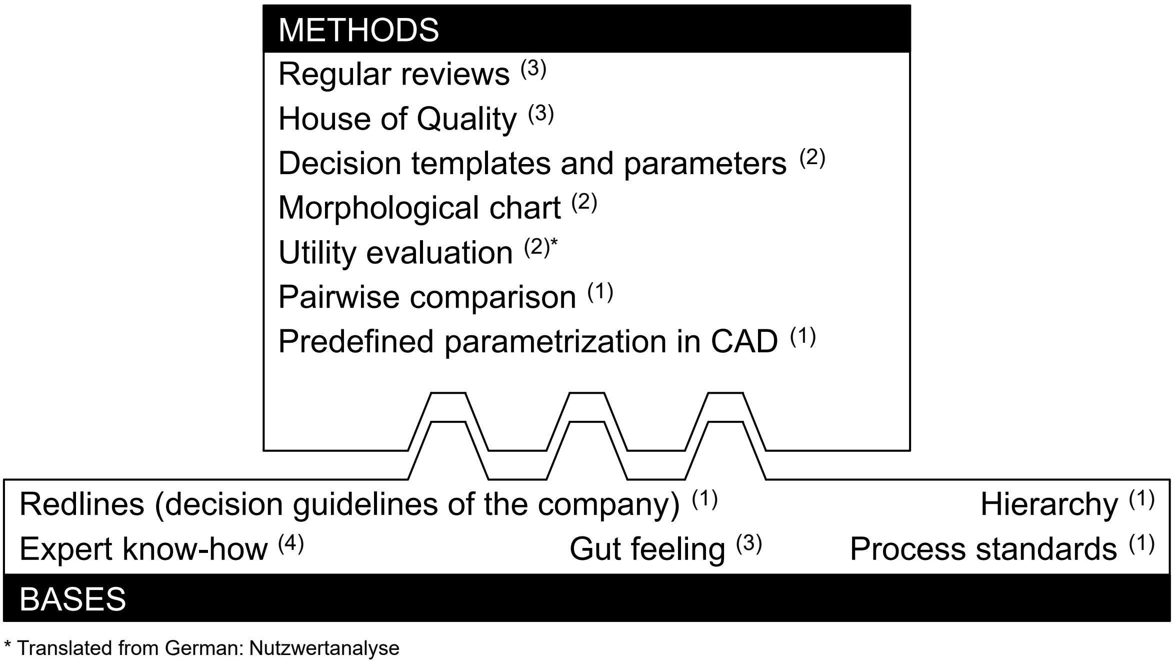


Figure 3.4: Methods and their bases used for making decisions in different companies. The number in parentheses indicates the number of mentions. Figure according to Stulpe (2020).

is the basis for decisions. This can lead to challenges, such as: lacking transparency (as subjective and individual measures form the bases of decisions), lacking technical in-depth knowledge, or long decision processes. Seven of the 17 interviewees named delays in the decision-making process due to the different views of the stakeholders and the lack of a holistic view. One interviewee considers the attempt to find a global optimum as very challenging. The consequences of decisions emerge much later in time. This makes decision-making complex.

Complexity

> "... what usually happens is that I make changes at one point and don't have all the consequences in mind, and afterwards it just bangs somewhere."[4]

Two of 17 interviewees did not mention complexity explicitly during the interview. Among the remaining 15, ten described product complexity as challenging, seven internal and six external complexity. Internal complexity is mostly related to communication and processes, such as decision-making. External complexity can arise from regulations, customers or different markets (globalization). The interviewees name the following causes for product complexity:

- increasing number of requirements
- increasing number of components
- increasing number of interfaces
- increasing external variety
- coupling of software and hardware
- coupling of mechanics and control
- increasing complexity of technologies
- dependencies between variables/components

[4] translated from German: "[...] was meistens passiert, ich ändere an einer Stelle und habe eben nicht alle Konsequenzen im Blick und hinterher knallts halt irgendwo." (INr. 8, Z.376f)

One of the main challenges which arises from product complexity is that the system behaviour becomes unpredictable after changing variable values or components. Managed product complexity is the reliable prediction of impacts on the overall system when variables, sub-systems or components have been changed. Only two participants state that they know the impact of changes on the overall product. The companies can master the behaviour of sub systems, but face problems when it comes to an holistic view. Increasing complexity leads to increasing costs. Nevertheless, managing complexity is difficult because circular dependencies complicate the identification of cause and effects.

Robust Design

> "Two keywords. First, simulation, really a lot of simulation. And second, very early tests in the known markets with high requirements."[5]

The interpretation of robust design differs between the companies. One wants to avoid changes due to changing requirements. A company developing high-tech electronics wants to include yet unexpressed requirements of the customer. They include interfaces or specify features during development because they anticipate what the customer will ask for in the end. A power tool manufacturer develops several alternatives during concept development to be able to react to changing requirements. A manufacturer of consumer goods interprets robustness for mass production: the manufacturer wants to avoid rejections during production because the tolerances are not met. A manufacturer of industrial vehicles sees robustness as ensuring safety, reliability and quality.
Companies use simulations and tests to ensure robustness. It is considered as especially useful in hardware development. One interviewee connects robustness with over-dimensioning, which can increase costs.

Variant Management

> "Product family design is definitely a main challenge for all products. Sometimes it has a certain logic. Sometimes it's just historically grown. There you hit a nerve because it's a current topic right now."[6]

External variety often makes variant management necessary. External variety can be caused by individual customer needs or market regulations. Sometimes customers directly influence standardization decisions. Companies also use variant management to quickly respond to market needs. The importance of variant management is strongly related to the company's context and the product: wide product portfolios and high production volumes often force companies to manage their product variants. An automotive company aims at reducing unnecessary variants within their components and strives for optimal standardization. An agricultural machinery company, a manufacturer of metal processing tools and a medical engineering company have historically grown portfolios. The degree of standardization has never been checked to be optimal; an overview is missing. A manufacturer of industrial vehicles needs to quickly respond to customer wishes and, thus, creates variants on demand. Designers work isolated and miss potential for standardization. Another company has to redesign their standardized component kit every two or three years. A power tool manufacturer struggles to fulfill all requirements of different product variants when it comes to modularization and standardization. Lacking variant management leads to:

- missing transparency and overview
- increasing risk of recalls
- high engineering effort for each order

[5] translated from German: "Zwei Stichworte. Erstens Simulation, richtig viel Simulation. Und zweitens sehr frühe Tests in den bekannten Märkten mit hohen Anforderungen." (INr. 7, Z.291ff)

[6] translated from German: "Die [Familienbildung] (d.V.) ist definitiv da, für alle Produkte. Manchmal hat sie eine gewisse Logik. Manchmal ist sie einfach historisch gewachsen. Da triffst du jetzt einen Nerv, weil das ist gerade aktuell. (INr. 5, Z.242)

Q1: How relevant do you evaluate the focus on the early phase? (from 1 - very low to 5 - very high)
H1: Automated model generation with modular models reduces the modelling effort.
H2: Product family design is a critical challenge for the future.
H3: In the future, quantitative analyses will significantly drive product development.

Interview	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	∑	Ø
Q1	4	5	4	4	5	4	5	5	5	4	5	5	3	5	5	5	73	4.6
H1	5	5	5	4	2	5	4	5	4	4	3	5	5	4	3	5	68	4.3
H2	4	3	3	5	3	5	3	3	5	3	4	5	3	4	4	3	60	3.8
H3	3	5	3	4	4	4	4	3	3	3	3	4	4	4	3	5	59	3.7
Company	N	L	L	M	J	A	E	B	D	I	C	G	H	O	F	K		

1	Strongly disagree
2	Disagree
3	Neither agree nor disagree
4	Agree
5	Strongly agree

Figure 3.5: Evaluation of hypotheses related to the research ideas. Figure according to Stulpe (2020).

- irritated customers due to indistinguishable product specifications and complex portfolios
- increased costs for storage
- increased internal efforts for documentation

3.2.3 Evaluation of Research Ideas

The interviewer presented the following two research ideas: The first research idea is about product family design. The product family design research idea is a quantitative approach. It is based on quantitative models, which describe the system behaviour. Algorithms identify the potential for standardization and commonality from a technical perspective within the product family. Among all technically possible commonalities, optimization algorithms seek for the cost optimum.

The second research idea is about automated model generation using modular models. Modular models are standalone models, which describe a certain aspect of the system at hand. An algorithm can automatically combine them to an overall system model. They are connected via standardized interfaces. Thus, complex models for product design can be divided into manageable parts.

The interviewees answered questions regarding the applicability, strengths, weaknesses and suggestions for improvement of the presented research ideas. A five-point Likert-scale was used to rate hypotheses (range: strongly disagree 1 - strongly agree 5).

After presenting these research ideas, some hypotheses were evaluated and the research ideas were discussed.

Hypotheses

Figure 3.5 shows how the interviewees rated their the relevance and agreement on a question and three hypotheses using a five-point Likert-scale (range: strongly disagree 1 - strongly agree 5). The interviewees highly agree that the focus on the early phase of the development process is highly relevant for product family design: 14 out of 16 rate it as highly relevant. It was rated with 4.6 on average. The early phase was described as the left part of the V-Model. The statement that an approach using modular models can reduce modelling effort was rated with 4.3 on average. None of the interviewees disagrees with the statement that product family design will be crucial challenge in the future, eight out of 16 agree or strongly agree (3.8 on average). This rating depends, as mentioned in section 3.2.2, on the products, which the company develops. The companies in this interview study,

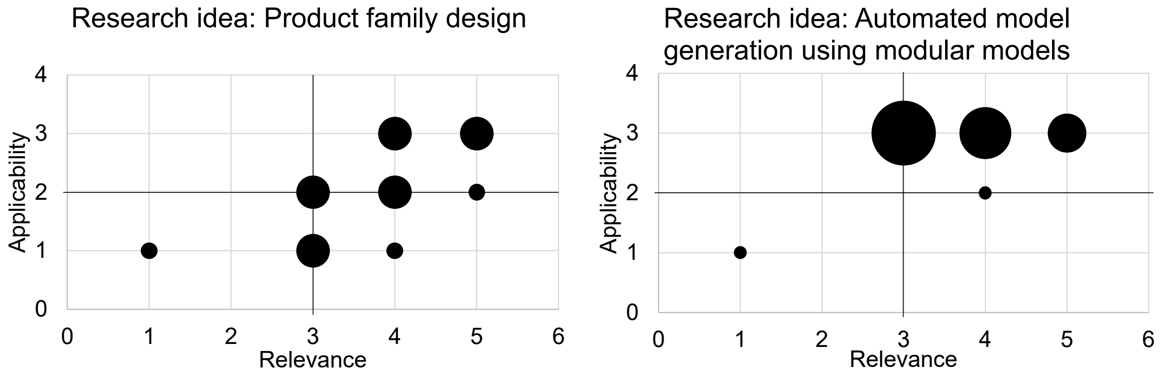


Figure 3.6: Rating of relevance and applicability of the research ideas. Left: product family design; right: modular models. The circle size indicates the number of mentions.

which produce a wide range of serial products, always work with some kind of standardization and, thus product family design is more relevant for them. None of the interviewees disagrees with the statement that quantitative analyses will strongly guide product development, nine out of 16 agree or strongly agree (3.7 on average). People with a medium ranking of three emphasize the cost-benefit ratio as decisive.

The interviewees were also asked to rate the relevance of the presented research ideas for their company on a five-point Likert-scale. They should give their level of agreement for the statement that the respective research idea is relevant for their company. During the interviews they were also asked to discuss the applicability of the research ideas. The student, who conducted the interview study, grouped those answers in three categories: low, middle, high with the numbers 1, 2, 3 respectively. Figure 3.6 depicts the results of the analysis. The bubble diameter indicates the number of answers. This depiction helps to summarize the results, it is not a statistical analysis. It contains the answers of those interviewees, of whom data for both dimensions is available (13 for product family design, 14 for modular models). In the interviews, which are not considered in Figure 3.6, data for at least one dimension was missing. One interviewee disagreed with both applicability and relevance of both research ideas for his company. It is a company providing software solutions. The interviewee considers both research ideas as unsuitable for this context. According to him, variant management in software development differs from classical variant management in hardware producing industry. Eight out of 13 agreed or strongly agreed that the research idea for product family design is relevant for their company; eight out of 14 did so for the research idea modular models. One person from an automotive company also mentioned that the importance of product family design will increase due to the increasing pressure on costs. She highlighted the design for commonality and searching for the cost optimum. For the applicability no tendency can be seen for the product family design research idea. But a higher rating for relevance goes along with an higher rating for applicability and vice versa. The research idea for modular models on the other hand is highly rated as applicable: twelve out of 14 think that this research idea could be applied in their company, rather independently from its relevance.

Research idea: product family design
This part summarizes the participants' answers regarding the potential and the concerns applying the product family design research idea. Participants consider the following points as potential benefits:

- reduction of unnecessary variant development
- faster decisions regarding variants and commonality
- reduction of development costs of product families
- identification of yet unknown potential for standardization/commonality

- decisions based on technical knowledge
- development of more robust systems
- reduction of iterations during development
- supporting standardized component and module design by optimization
- reducing internal variety while external variety is increasing

Thus, the participants consider the research idea as a support for decision-making and variant development.

The following statements summarize concerns and barriers in applying the product family design research idea.

- availability of relevant information, such as requirements
- uncertainty of cost predictions, especially of subsequent costs
- decisions are not only technically based, but also on soft factors, such as the company's policy
- variants, which emerge later, have to be considered as well
- the user must be well acquainted with the product family design approach, which can decrease the willingness to use it
- market prices for components can change and influence the results

Also a general aspect of product family design was mentioned: the change effort in product families is higher than for one product. Components cannot be simply changed, due to interactions with other products of the product family. This leads to an increased complexity. One interviewee describes the research idea of product family design as helpful and very practical, but states that it should be used in consulting rather than product development. Direct handling with MatLab™ code should be avoided by a graphical user interface. Using the tools should not add additional complexity.

In order to cost-optimize the product family, the algorithm needs a cost model as an objective function. Nine interviewees out of 15 companies said that they can access cost models. Three interviewees said that they do not have access to cost models. The remaining three participants could not give a statement. The interviewees highlight the uncertainty of cost models, especially when it comes to complexity costs. One interviewee considered the effort in modelling those costs as too high.

Research idea: automated model generation with modular models

This part summarizes the participants' answers regarding the potential and the concerns of the research idea on automated model generation using modular models. The following list summarizes the benefits named by the interviewees:

- transparent documentation of the explicit and implicit knowledge about the product throughout the development process
- accessability for all parties involved
- re-usability through modularization and pre-validation of modular (sub-)models
- easy comparison of different concepts without prototyping effort
- reduced development effort for multiple application of models
- higher performance and lower effort compared to manual modelling of dependencies
- developers are forced to deal with the structure of the product and to clearly formulate requirements.
- clear visualizations show change effort and change propagation
- cause-and-effect structure supports risk management
- identification of re-occurring requirements and their associated solution concepts.
- all requirements and dependencies are documented and, thus, cannot be forgotten
- support for robust design
- making complexity manageable
- supports model-based decision-making

Knowledge about cause and effect is in the minds of different designers. Bringing this knowledge together and presenting it in a transparent and easily understandable way for everyone is the most frequently cited advantage of using a modular database. The participants emphasize the reduction of effort by re-using verified models, transparent documentation of knowledge, and clear visualizations. One thinks, that it can enable collaboration within and between departments. System architects of a producer of high-end electronics are looking for a tool to store multiple concepts in one place. A interviewee from an automotive company highlights the holistic character of the idea. Whereas models for sub systems are available, there is no holistic view on the product. The following list summarizes the participants' concerns in applying automated model generation with modular models.

- modelling and combining the existing can impede searching for completely new ideas and innovations
- high initial effort for modelling the system dependencies
- 'shit in, shit out' [7]: bad models can influence the results negatively for a long time.
- high effort for maintenance
- in early phases the product architecture might not be known
- missing interface to CAD and PDM systems, risk of parallel structures
- should be easy to use
- integration of non quantifiable requirements unclear/ not possible
- only advantageous if applied on the long term
- effort for keeping data up-to-date
- computational effort should not be too high

A database, where all the models are stored, should be generic enough to be applicable in different industries. One interviewee had doubts when documenting complicated interdependencies and solving coupled problems. Furthermore, a user should be able to integrate binary variables. There is also an bureaucratic obstacle in some companies: the benefit of applying this research idea comes later than the effort. Thus, it could be difficult to argue for the research idea. One persons states that quantification of the dependencies is just a matter of time, as soon the dependencies are clear.

[7](INr. 11, Z.534)

4 Requirements of a New Approach

This chapter summarizes findings from literature (see chapter 2) and empirical studies (see chapter 3). It formulates requirements of the developed approach. There are three sources for requirements: (1) industry-related challenges, (2) academia-related challenges, and the (3) evaluation of the research ideas.

4.1 Industry-related Challenges

The empirical studies in chapter 3 revealed general challenges that a wide variety of companies are facing. This section summarizes insights which are important for the approach to be developed.

The **early phase** of the product development process is often crucial for the success of products. In many companies the V-model provides guidance. However, managing **requirements** challenges most of the companies. Conflicts of goals appear in nearly all companies. Those conflicts of goals delay development times or can even lead to a restart of a development project. Products need to fulfill requirements from different disciplines and sources. Therefore, a multi-disciplinary view of the product and its requirements is necessary. Companies also aim to quantify requirements and to objectify **decision-making**. Although some interviewees mention methods to support decision-making, the basis of decision-making is mostly dominated by gut feeling, expert opinion or simply hierarchy. Different point of views combined with a lacking multi-disciplinary view lead to exhaustive discussions and might delay decision processes. Some participants highlight the need for a technical basis and transparency. Challenges in decision-making are often connected to **complexity**. The complex structure of products complicates the prediction of the emergent system behaviour and, thus, the impact of decisions. Changing one component can influence other components. Standardizing one component can lead to increased variety of other components. Many companies can handle sub-systems, but they struggle when taking the whole system into account. Cause and effect relations become unclear. Increasing number of components, interfaces and dependencies lead to higher product complexity. Higher complexity leads to higher costs. At the same time companies have to face an increasing cost-pressure. Some companies introduce **variant management** to reduce cost. However, market needs often generate variants without control. The portfolio grows historically with high numbers of variants for one product. Isolated and incomplete views of the product and complex dependencies make it difficult to standardize or even optimize portfolios. This leads to lacking transparency, increasing costs, and irritated customers when portfolios are complex and products hard to distinguish. Standardization by over-dimensioning can reduce component variants, but at the same time increase manufacturing costs. This makes it difficult to find an optimal solution. One interviewee mentions that over-dimensioning can also enable **robust design**. Companies define robust design very differently. One interpretation is the robustness against changes in requirements, which is seen especially useful in hardware development. Across all topics, the interviewees ask for methodical support.

4.2 Academia-related Challenges

Chapter 2 presents current research of product family design. This section summarizes the challenges deducted from literature.

Optimization problems which optimize both the assignment scheme and the component design are called class III problems. Solving a class III problem with generalized commonality is computationally challenging. On the one hand, it is a mixed integer non linear problem. On the other hand, the number of possible combinations to share component variants in a product family without restrictions increases significantly with the number of product variants and components. This leads to high computational effort for t'classical' optimization algorithms, as presented in section 2.3. Furthermore, ensuring commonality and interfaces within the components on the level of the design variables is challenging. There is a need for a **fast calculating class III algorithm with generalized commonality**.

Finding an **objective function**, which includes all relevant variables, is still a challenge. Some metrics include hidden assumptions or simplify the trade off of standardization to maximal commonality. The goal here is to define a objective function, which depicts the relevant influences of standardization directly without assumptions or decision-making (pareto front).

4.3 Evaluation of Research Ideas

Both research ideas for product family design and automated model generation using modular models were mostly rated as relevant. However, the interviewees have doubts about the applicability of the product family design approach. In contrast, they rated the applicability of the modular models approach as high.

The interviewees think, that the **product family design** approach can help to identify standardization potential, reduce unnecessary variants and overall costs. It improves decisions by basing them on technical knowledge. They have doubts because of the uncertainty of available data, especially complexity costs or changing prices for components. They think that the user should be able to interact with the tool to take soft factors into account. At the same time, the user should be able to use the tool without going to deeply into coding.

Automated model generation using modular models can serve as transparent database for knowledge management, if it is accessible for all parties involved. It can document explicit and implicit knowledge about the product: all dependencies and requirements are documented. The causal structure can avoid confusion of cause with effect. It can visualize change propagation from components to the overall system. Pre-validated modular models can reduce modelling efforts and make complexity manageable. It can assemble existing sub system models to a holistic model. On the other hand, concentrating on existing solutions can impede innovations. Also the initial effort to set up the database and maintenance can be high. The product's architecture might not be known yet. It is not clear how to include non-quantifiable requirements. Bad models can lead to bad results. It should generic enough to support different industries.

4.4 Requirements for a New Approach

This section summarizes requirements from the summaries of the previous sections. The goal is to develop a novel approach for product family design based on the current needs and challenges of the industry. The requirements for the approach reflect its goal (Blessing and Chakrabarti, 2009).

Goal
Develop an approach to support designers in optimizing product families subject to complexity and uncertainty.
Functional Requirements
Optimizes class III problems with generalized commonality.
Optimizes components with more than one design variable.
Applicable during the early phases of the development process.
Includes requirements of the product variants.
Supports resolving conflicts of goals.
Allows to incorporate different disciplines.
Provides transparency during complex system design.
The objective function incorporates relevant effects of standardization.
Provides visualizations for decision support in product family design.
Allows for robust design of product families.[a]
Application Requirements
Guides and supports the user during application.
Applicable independently of problem and user.
Faster than existing approaches.

[a] Robustness in this context refers to technical and economical robustness. Technical robustness tolerates changing component attributes. Economical robustness tolerates changing predictions of costs and production volumes.

Table 4.1: Requirements of a new approach.

5 Fundamentals of the Developed Approach

This chapter provides the reader with relevant background information on methods and approaches which are used in the developed approach. Reading of this chapter is only necessary if the reader is not familiar with the following sections: 5.1 Process Design Structure Matrices, 5.2 Solution Space Engineering, 5.3 Maximizing Commonality Using Solution Spaces, or 5.4 Optimization Algorithms.

5.1 Process Design Structure Matrices

Eppinger and Browning (2012) introduce the concept of design structure matrices (DSMs) and give an overview of application examples. DSMs can represent a static architecture, such as a product or an organizational architecture, or a temporal flow, such as processes.

Process DSMs are usually used for modelling and improving processes; in particular for product development processes (Eppinger and Browning, 2012).

A process DSM has activities as elements and dependencies as nodes. Dependency in this context means that an activity receives information from another activity. The matrix (DSM) and its graphical representation (graph) contain the same information and are exchangeable.

Figure 5.1 shows both the graph and the corresponding DSM. It is a directed graph. Thus, the DSM is not symmetric. This work follows the so called input in columns/ feedback below diagonal (IC/FBD) convention. This means that the flow of information is from column to rows. In this example, the orange arrow indicates the flow of information from C to B. The '1' means that C provides input for B. Entries below the diagonal indicate activities that are executed, although they depend on inputs from downstream activities. Here, B is executed before C, but C provides input for B.

The 'sequencing' method reorders rows and columns so that ideally all entries are above the diagonal. This means that activities can be executed sequentially or in parallel without feedback. Sometimes activities are coupled. In this case sequencing tries to make the feedback loops as small as possible, i.e., as closest to the diagonal as possible. Here, moving C to the top changes the execution order. C is now the first activity, followed by A and B. A & C can be executed sequentially or in parallel. There is no feedback left.

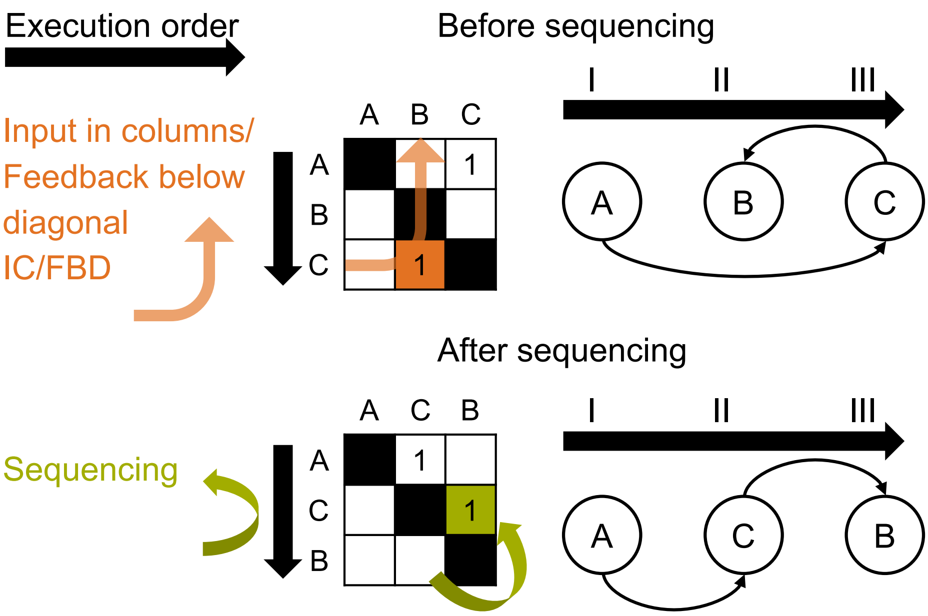


Figure 5.1: Temporal process DSM (middle) with corresponding graph (right) before (top) and after sequencing (bottom) according to IC/FBD convention.

5.2 Solution Space Engineering

Zimmermann et al. (2017) introduced the approach of solution space engineering (SSE). It can be seen as the quantification of the left side of the V-model. Starting from requirements on the system level, SSE breaks down the requirements to the sub-system or component level.

Figure 5.2 shows the three pillars of SSE. At first the dependencies are modelled using a dependency graph. It distinguishes between the level of design variables (DVs) at the bottom and the level of quantities of interests (QoIs) at the top. DVs represent components or sub-system. QoIs represent the emerging system behaviour. Bottom-up mappings quantify the dependencies from the level of design variables (bottom) to the level of quantities of interest (top). Top-down mappings, such as solution spaces, use the bottom-up mappings to break down the requirements from the system to the component level such that all requirements on the system level are fulfilled.

n design variables constitute an n-dimensional design space. A design is one point in that n-dimensional design space. A design may assume specific design variable values. A Monte Carlo sampling generates these design points randomly. The bottom-up mappings evaluate the system response for each design. This results in specific values of quantities of interest. Then the algorithm checks whether the values of the quantities of interest fulfill all requirements (Zimmermann and von Hoessle, 2013). Designs, which fulfill all requirements are called good designs.

The set of all good design points constitutes the complete solution space. It is a subset of the design space. Two dimensional projections of the n-dimensional design space can visualize the design and solution space.

Figure 5.3 shows two dimensional projections of a six dimensional design space. Each diagram depicts two design variables. The first row of the figure visualizes the complete design space. Good solutions (green dots) are hardly visible. Most design points have the colour of the requirement they violate. By narrowing down the box-shaped solution space the user can exclude bad designs. As these are projections of a six dimensional design space, a change in one diagram also influences the other diagrams. The user navigates through the design space by moving the limits of the box-shaped

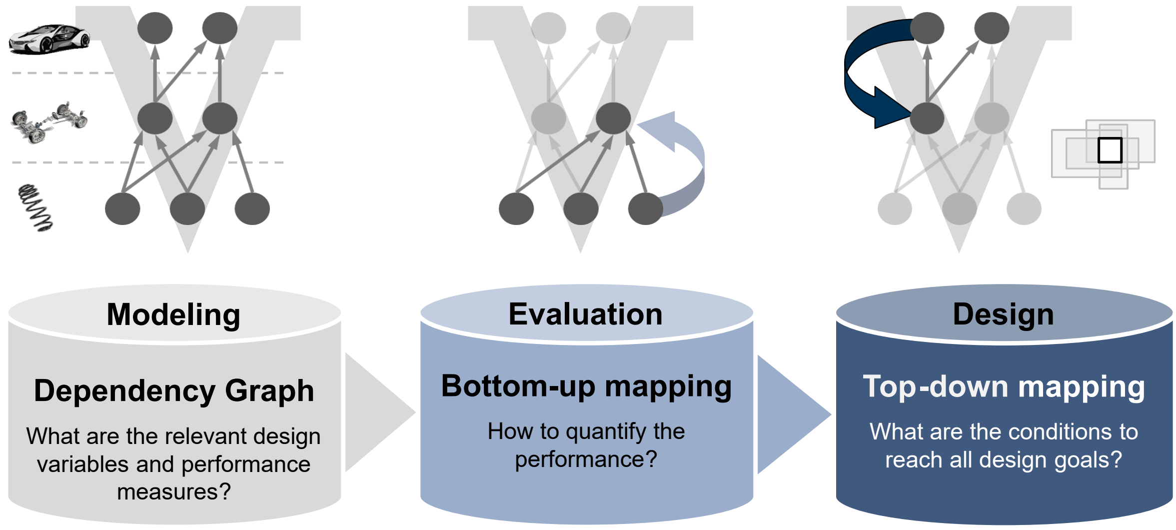


Figure 5.2: Three pillars of solution space engineering: dependency modeling, bottom-up mapping and top-down mapping. Figure according to Zimmermann et al. (2017)

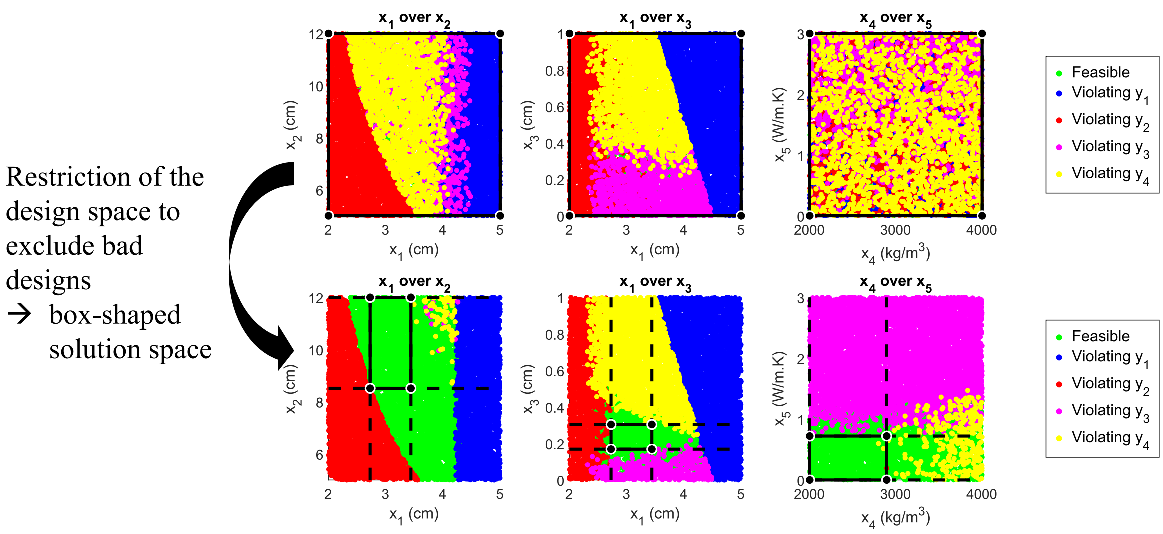


Figure 5.3: 2D projections of a six dimensional design space with box-shaped solution spaces. Top: initial situation with complete design space as candidate space. Bottom: box-shaped solution space containing only good designs

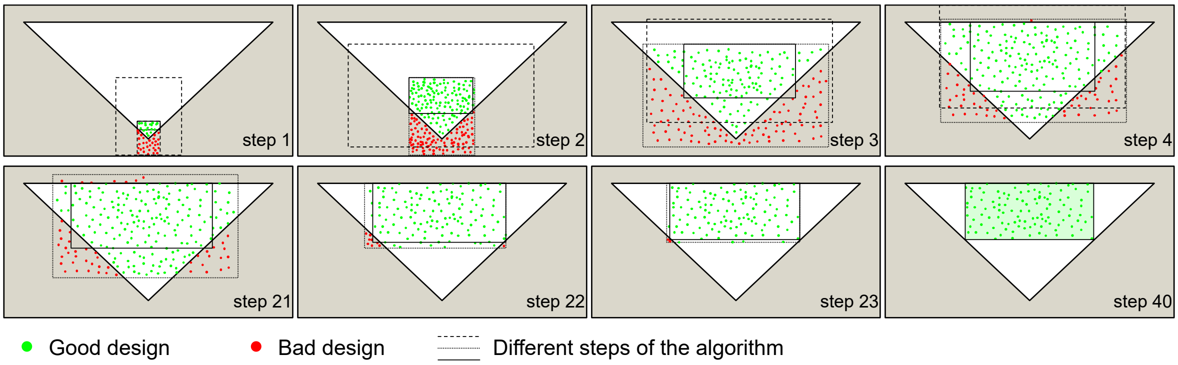


Figure 5.4: Algorithm to maximize box-shaped solution spaces. Figure according to Zimmermann and von Hoessle (2013)

solutions space. The goal is to end up with a box-shaped solution space which only contains green dots, i.e., good designs.

The box-shaped solution space is a subset of the complete solution space. On the one hand, it does not contain all good designs. On the other hand, it decouples the design variables from each other. The box-shaped solution space directly define intervals for each design variable. Within these intervals all requirements are fulfilled. Due to the box-shaped form each design variable can vary its value independently from the other design variables and it will lead to a good design – as long as all design variable values stay in their respective intervals. These intervals are the requirements for the design variables which are broken down from the system level.

Box-shaped solution spaces can be determined manually or by algorithms. Zimmermann and von Hoessle (2013) and Graff et al. (2016) propose an algorithm to find maximized solution spaces. The basic idea is that maximized solution spaces allow for maximal flexibility during design. Designers can move freely within the intervals of the box-shaped solution space. The bigger it is, the more freedom a designer has to adapt and change a design.

The algorithm searching for the largest box-shaped solution space has two phases: (I) an expansion phase where the algorithm tries to increase the number of good designs included in the box-shaped. (II) an consolidation phase where the algorithm removes bad designs by making the box-shaped smaller. Figure 5.4 illustrates the functionality of the algorithm. In the following the terms box-shaped solution space and solution space are used synonymously.

5.3 Maximizing Commonality Using Solution Spaces

Eichstetter et al. (2015) proposed an approach to maximize commonality in product family design. Figure 5.5 illustrates how solution spaces can be used to identify commonality. Solution spaces define the permissible intervals of a design variable for a specific product A. Same applies for further products, such as product B. Design variables describe components. The approach uses the algorithm to maximize solution spaces for each product from section 5.2. In the approach by Eichstetter et al. (2015), each component is described by one design variable. Here, x_1 represents component 1 and x_2 represents component 2. The solution spaces of both products A & B overlap for both design variables. This means product A and product B can share variants of component 1 and component 2. In general: If the solution spaces of a design variable for different product variants overlap, there are design variable values which fulfill the requirements of both or more products. Thus, the product

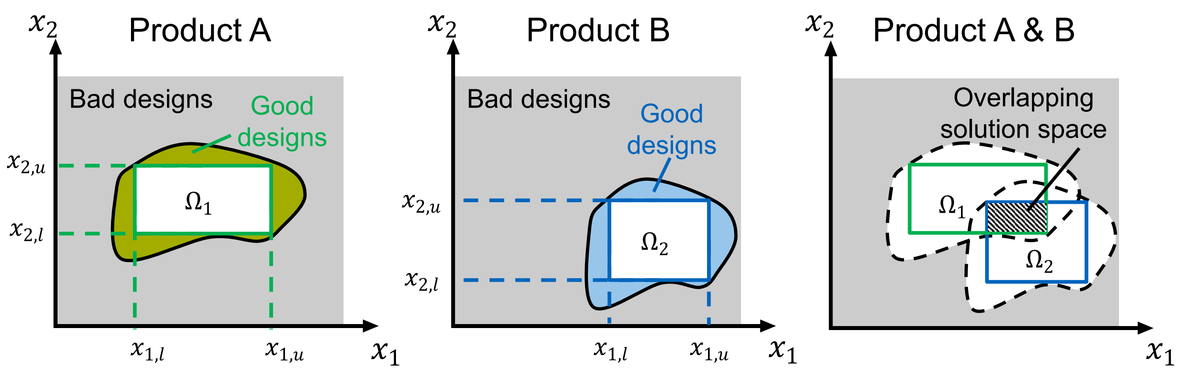


Figure 5.5: Using solution spaces to identify commonality between components of different products. Figure according to Eichstetter et al. (2015)

variants can share the component variant.

Then the algorithm iterates over all design variables. One DV describes one component. The algorithm minimizes the number of components needed to realize all products. It starts with maximum commonality, i.e., only one component variant for all products and checks whether there is a common solution space. If not, it increases the number of component variants and checks again until there is a common solution space. This means that this assignment scheme is feasible and the algorithm stops.

5.4 Optimization Algorithms

Equation 5.1 shows general problem statement of an optimization algorithm (Papalambros and Wilde, 2017).

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \textit{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u \end{aligned} \tag{5.1}$$

Optimization algorithms search for optimized values $\mathbf{x}^*$ of the design variables $\mathbf{x}$ which minimize the objective function $f(\mathbf{x})$. The design variables are usually subject to constraints. $\mathbf{g}(\mathbf{x})$ is the vector of inequality constraint functions. $\mathbf{h}(\mathbf{x})$ is the vector of equality constraint functions. The lower and upper bounds of $\mathbf{x}$, $\mathbf{x}^l$ and $\mathbf{x}^u$, limit the design space where the algorithm searches for the optimum.

Papalambros and Wilde (2017) distinguish between gradient and nongradient methods or algorithms of optimization. Gradient methods use slope or curvature information to identify optimal solutions. They require continuous functions and the optimum found depends on the starting point if the function is not convex. Nongradient methods use only the function values themselves and have no requirements on the functions they use.

The approach presented in this thesis uses one gradient method (interior point algorithm) and three nongradient methods (exhaustive search, genetic algorithm and particle swarm optimization).

The **interior point algorithm** used in this thesis solves a non linear optimization problem without

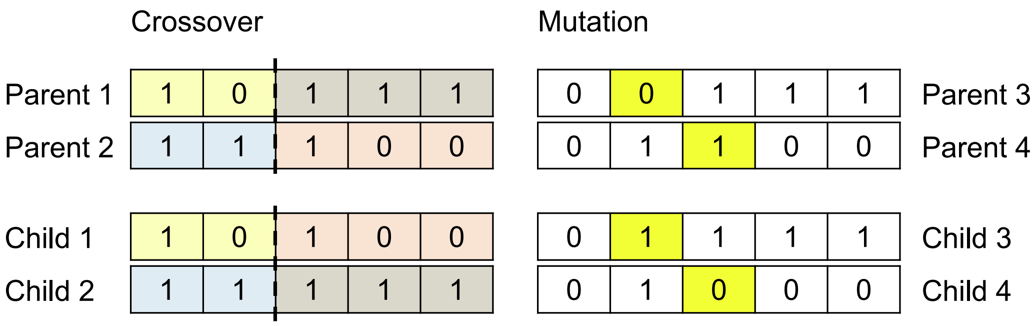


Figure 5.6: Example for crossover and mutation in genetic algorithms

constraints. It uses gradient information to find the optimized solution: It uses the Hessian matrix to define the search direction and stops the optimization when the so-called first-order optimality condition is satisfied: $\nabla f(\mathbf{x}^*) = 0$. The algorithm used is *fmincon* (MATLAB, 2020).

Exhaustive search calculates all possible solutions of the optimization problem and compares the objective function values to find the smallest. Usually the number of possibilities is infinite as design variable values are real numbers. But for smaller combinatorial problems exhaustive search can reliably identify the optimal solution.

When the number of combinations becomes too big, there is a need for more sophisticated methods. **Genetic algorithms** (GAs) imitate the evolutionary process. A chromosome represents a design point. It is encoded as binary values. Thus, GAs are suitable for discrete or binary design variables (Papalambros and Wilde, 2017). A set of design points or chromosomes constitute a population.

The so called fitness function determines the quality of a chromosome. It is the objective function of the optimization problem. The general idea is that chromosomes with higher fitness function values have a higher chance for reproduction than the ones with lower values. Therefore parents with high fitness function values are selected. Those chromosomes have two possibilities for reproduction: crossover and mutation.

Figure 5.6 illustrates the two main methods for reproduction. During crossover, the chromosomes of the parents are cut at a defined position and then recombined. Here, child 1 receives the first segment from parent 1 and the second segment from parent 2. Child 2 receives the first segment from parent 2 and the second segment from parent 1. Mutation generates a new child from one parent by switching one or more entries of the parent's chromosome randomly. In this binary example, mutation of the second entry of the chromosome of parent 3 produces child 3. Child 4 was produced similarly with entry three of parent 4. Sometimes the best parents are saved directly to the new generation without changes to keep them. They are called elite.

The children constitute the new generation. The fitness function values of the new generation is evaluated and the process described above starts again. This is done until the convergence criteria is satisfied. This can be a certain number of iterations or the improvement of fitness function values falling below a minimum level of improvement. Figure 5.7 (left) shows the flow chart of a GA with the steps described above.

Particle swarm optimization (PSO) mimicks the behaviour of swarms, such as insects or birds. The main difference to GAs is that populations in PSO follow a trajectory whereas populations in GAs jump in the design space (Papalambros and Wilde, 2017).

Each particle in a PSO population has its own position and velocity in the design space. At first, the position of the particles is determined. If it is better than the previous individual best position it becomes the new individual best position. Same applies for the best position of the whole swarm,

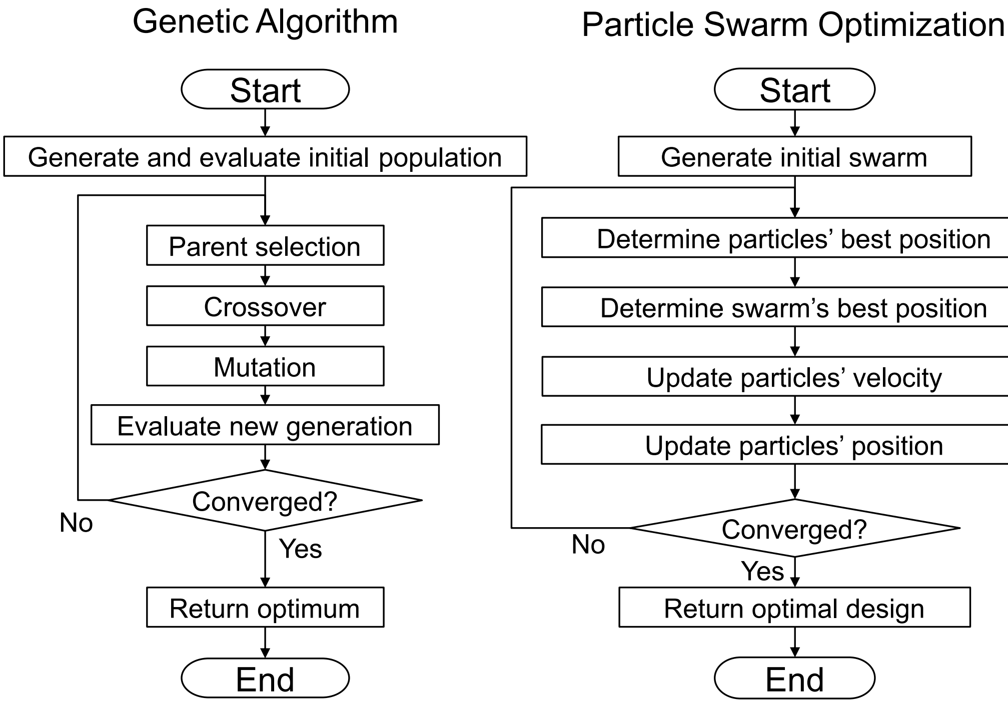


Figure 5.7: Flow charts of a genetic algorithm (left) and a partical swarm optimization (right) (figure according to Fujita and Yoshida (2004) (GA) and Kennedy and Russell Eberhart (1995) (PSO))

which is the best individual value. After determining the best positions of the particles and the swarm, the velocity for each particle is calculated. The idea is that a particle moves between its individual best position $\mathbf{p}_i$ and the swarm's best position $\mathbf{p}_g$. By updating both the swarm keeps moving towards better objective function values and exploring the design space around the population. Equations (5.2) and (5.3) update velocity and position of the particles.

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{u}(\phi_1). * (\mathbf{p}_i - \mathbf{x}_i) + \mathbf{u}(\phi_2). * (\mathbf{p}_g - \mathbf{x}_i) \tag{5.2}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_i \tag{5.3}$$

$\mathbf{u}(\phi_1)$ and $\mathbf{u}(\phi_2)$ are vectors of independent random variables distributed within $[-\phi_1; \phi_1]$ and $[-\phi_2; \phi_2]$ respectively. They are multiplied element-wise $(.*)$. The parameters ϕ_1 and ϕ_2 determine how fast a particle moves within a design space. Small values lead to a more stable algorithm but it is possible that the swarm cannot find the global optimum. Large values lead to a greater exploration of the design space (global search), but also increase convergence difficulties (Papalambros and Wilde, 2017).

Updating the swarm is repeated until a convergence criterion is met. This can be again a certain number of iterations or the improvement of the best position falling below a minimum level of improvement. Figure 5.7 (right) shows the flow chart of a PSO with the steps described above.

Both algorithms GA and PSO do not scan the whole design space. They have stochastic elements, such as mutation or the random distribution of $\phi_{1,2}$. This can lead to different results when optimizing the same problem again, which then can limit the reproducibility of the optimization results.

6 Developed Approach

This chapter presents the developed approach. Section 6.1 gives an overview of the steps which need to be conducted and their outputs. Sections 6.3 to 6.7 describe the methods used within the steps. The steps are explained using a simple transmission example. This example is introduced in section 6.2

6.1 Overview and Prerequisites of the Developed Approach

The goal of the developed approach is to find a cost-optimized product family design. The optimized product family design includes an optimized assignment scheme and optimized design variable values of the components. Both optimizations are solved in a joint problem. The approach incorporates both a technical and an economical point of view.

Figure 6.1 shows the procedure model of the developed approach. It consists of five steps with specific outputs. The approach starts with collecting the requirements which represent the product variants. Step 2 models the dependencies of the system. It aims to generate an understanding of cause and effects between the design variables of the components and the quantities of interest of the system. Step 3 quantifies the dependencies from step 2. The basic idea is to split the complex system into manageable pieces and model those separately. A database stores the models. They are then combined to the overall system model automatically. These system models quantify the technical behaviour of the system. Step 4 incorporates the economical perspective by modelling the cost of the product family. The resulting cost model serves as an objective function for the cost optimization of the product family in step 5. Here the user can choose between two optimization algorithms: a rather classical optimization approach based on algorithms of the literature research (see chapter 2) or a new method using solution spaces. In both cases the results are an optimized assignment scheme and optimized design variable values of the components. The approach is not purely sequential. The user can jump back and forth to complete the activities.

Methods from step 2 - 5 of the procedure have been published. Table 6.1 gives an overview of the references.

Step	Activity	Reference
2	Model System Dependencies	Rötzer et al. (2022b)
3	Model System Behaviour	Rötzer et al. (2020a)
4	Model Cost	Ehrlenspiel et al. (2020); Rötzer et al. (2022a)
5	Optimize Product Family	Rötzer et al. (2020b); Rötzer et al. (2021)

Table 6.1: References for the elements of the developed approach

The approach is a quantitative approach. This comes along with certain prerequisites. Table 6.2 gives an overview of those prerequisites.

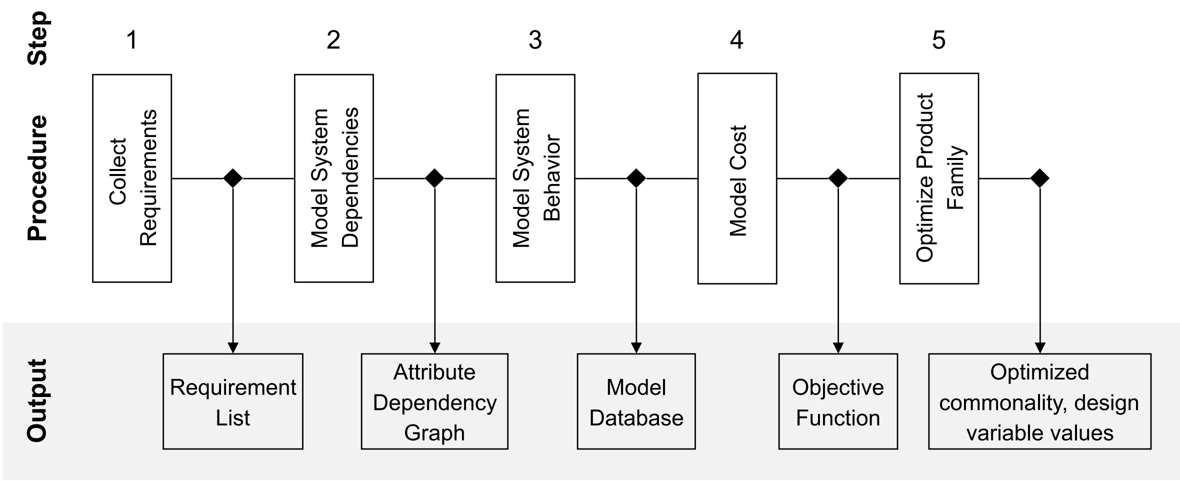


Figure 6.1: Steps and outputs of the developed approach

ID	Prerequisite
PR 1	Scale-based product families, i.e., scaling of design variables generates new component and product variants.
PR 2	Requirements on the product need to be formulated quantitatively.
PR 3	Product variants differ from each other by different values for their requirements.
PR 4	Design variables are quantifiable, i.e., the component variants differ from each other by values of design variables and not by architecture.
PR 5	System behaviour is quantifiable, i.e., the design follows a quantitative procedure, such as testing, simulations, or formulae.

Table 6.2: Prerequisites of the developed approach

6.2 Explanatory Example: Product Family Design of a Simple Transmission

This section introduces a simplified transmission example. This example is used to explain the steps of the procedure. Figure 6.2 sketches a two stage transmission. Transmissions convert incoming torque T_{in} and number of revolutions n_{in} to an output torque t_{out} and an output number of revolutions n_{out}. Gears with different numbers of cog teeth ($z_{11}, z_{12}, z_{21}, z_{22}$) generate different transmission ratios (i_1,i_2) when being combined. These transmission ratios determine how the incoming torque and number of revolutions are converted. The first stage consists of spur gears the second of worm gears.

The transmissions in this example are used for screw conveyors in various industrial applications. They are located between the electric motor and the screw conveyor. Different applications and different electric motors require different product variants. The design task is: How should the gears be distributed within the product family? What should the components look like?

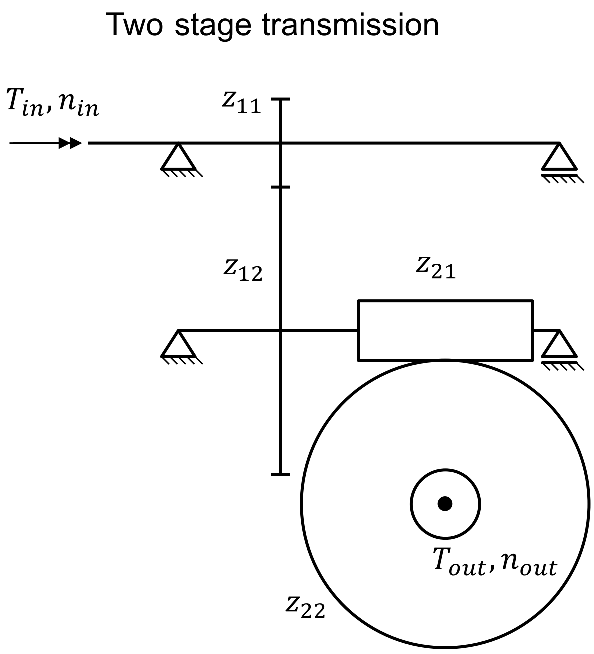


Figure 6.2: Sketch of a two stage transmission. First stage spur gear, second stage worm gear

6.3 Step 1: Collect Requirements

In the first step the requirements of the different product variants need to be collected. In this case different electric motors and different use cases lead to different product variants. Table 4.1 gives an overview of the product variants which should be offered and lists their requirements.

Soft factors can influence the decision on how to position products on the market, such as the perception of a brand, the transformation of business models, or other strategic motifs. Also the buying behaviour of customers can be subject to irrational behaviour. This makes the customer side (front-end) hardly quantifiable and predictable. Therefore, the developed approach does not optimize external variety, i.e., which products to offer on the market. The product variants are to be defined first. The approach then optimizes the internal variety.

Benchmarking compares products systematically according to specific attributes (Lindemann, 2016). It can help to define market segments and derive requirements of product variants. Those requirements are often used to distinguish the products on the market from each other. Therefore, they need to be

Electric motor	I		II
T_{in}	$7Nm$		$9Nm$
n_{in}	$314\frac{rad}{s}$		$392\frac{rad}{s}$
Product variants	1	2	3
T_{out}	$\geq 620Nm$	$\geq 700Nm$	$\geq 1500Nm$
n_{out}	$\geq 1.6\frac{rad}{s}$	$\geq 1.5\frac{rad}{s}$	$\geq 1.3\frac{rad}{s}$

Table 6.3: Product variants and their requirements for the two-level transmission example

communicated to customers and be accessible.

The result is a requirement list for the different product variants (see Table 6.3).

6.4 Step 2: Model System Dependencies

Illustrations of dependencies can support complex system design. Attribute Dependency Graphs (ADGs) can model system behaviour without circular dependencies Rötzer et al. (2022b).

Figure 6.3 shows the ADG of the simple transmission problem. On the left the abstracted ADG provides an overview of the attributes involved. The detailed ADG on the right allows for in-depth analysis of the system.

Attributes at the bottom of the ADG can be variable (design variables) or predetermined (design parameters). On this level a designer can influence the system by changing the values of the design variables (DVs). As soon as a designer assigns values to the DVs, the attributes on the upper levels are determined. Quantities of interests (QoIs) are the attributes on the highest level. They represent the system behaviour. The flow of information is always from the design variables to the quantities of interest (bottom-up).

In this case, the design variables are the cog teeth of the gears. A designer can directly influence these variables. Input torque and input revolutions are also on the lowest level as they are also determining the system behaviour at the top. But in contrast to the cog teeth, those attributes are fixed by the predefined electric engines in Table 6.3. Thus, they are depicted as design parameters with dashed lines. As soon as z_{11} and z_{12} are assigned values the transmission ratio i_1 is specified. Same applies for i_2 respectively. i_1 and i_2 determine i. i and T_{in} determine T_{out}. i and n_{in} determine n_{out}. So far, no requirements are introduced. ADGs represent emerging behaviour instead of desired or required behaviour. In this context, design is the process of assigning values to design variables such that the emerging quantities of interests fulfill all requirements of the product. Furthermore, attributes on the same hierarchical level are independent from each other, i.e., z_{11}, z_{12}, z_{21}, z_{22} can be assigned values independently from each other. This means that the design of z_{11} does not influence the design of z_{12} or vice versa. Figure 6.4 summarizes the rules for building ADGs.

The procedure, shown in Figure 6.5, helps to build ADGs. It consists of five steps. Step 1: the system boundary needs to be defined. In product family design this can be an entire product or also modules of a product. As this work presents a scale-based approach, it cannot be applied to configurable products. In that cases it makes sense to select certain modules which can be designed by the approach.

Step 2: The user collects the quantities of interest. Those attributes describe the system behaviour and are often constraint by requirements. Thus, they can be found in requirement lists or also in material used for communication with the customer (e.g., websites).

Step 3: Designers can think of attributes they usually change or need to determine during the design process. Those attributes constitute the design variables (variable) and design parameters (predetermined). In this step also intermediate attributes, which are neither design variables nor quantities of interest, may appear.

Step 4: Users re-arrange the attributes collected before. They group them on hierarchical levels. Several levels between design variables and quantities of interest are possible. Intermediate attributes on the levels between DVs and QoIs help to structure the graph and foster understanding and transparency.

Step 5: After specifying the hierarchical levels of te attributes the users can start to determine the dependencies. They always point from the bottom to the top. Experience has shown that starting from

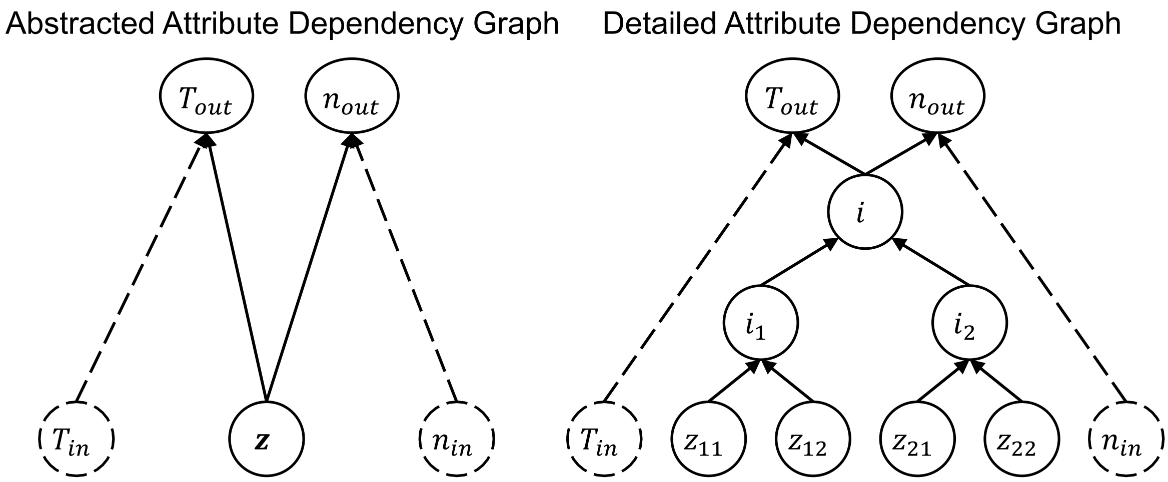


Figure 6.3: Dependency structure of the two-level transmission example. Left: abstracted attribute dependency graph (ADG); right: detailed ADG

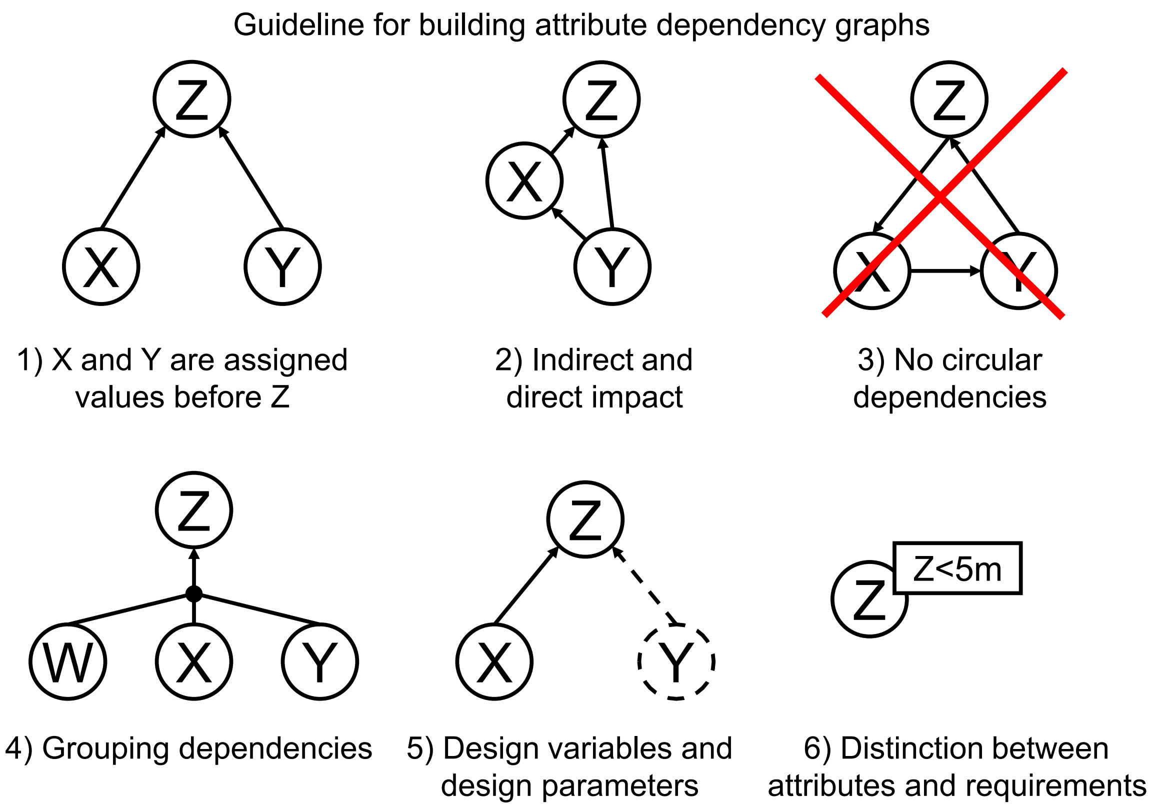


Figure 6.4: Rules for building ADGs. Figure according to Rötzer et al. (2022b)

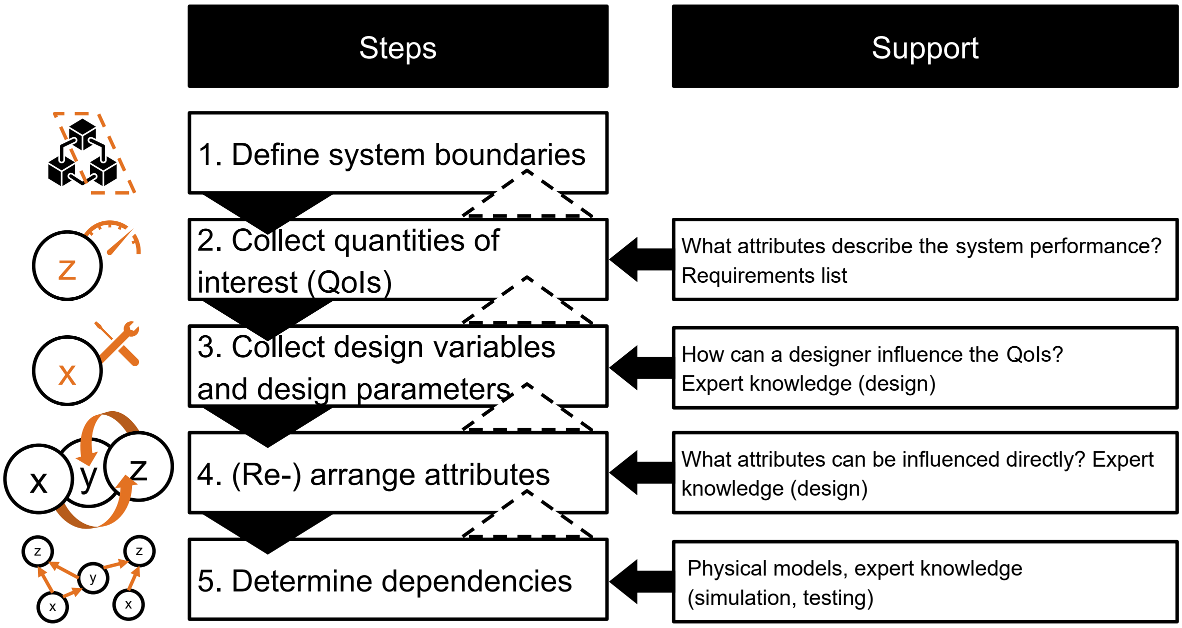


Figure 6.5: Procedure for building ADGs. Figure according to Rötzer et al. (2022b)

the level of QoIs works well. One can ask the following question: Which attributes determine the behaviour of this QoI? Then they can go through attributes on the subordinate levels and draw arrows from those attributes to the QoI. This can be repeated with all attributes until reaching the level of design variables.

The procedure is not linear as indicated by the dashed arrows. Steps can be repeated or a user can go back when identifying new attributes. The process is usually iterative until all relevant attributes are considered.

6.5 Step 3: Model System Behavior

After establishing the dependency structure by ADGs, the system needs to be modeled quantitatively. This means that for all arrows of the ADG quantitative models are to be provided. Quantitative models in solution space engineering are also called bottom-up mappings as they quantify the emerging system behaviour from the level of DVs to the level of QoIs. They are needed for quantitative analysis in section 6.7.

This approach uses a model database which combines modular sub-models of the system automatically. This model database with modular models is introduced to overcome the obstacles in the application of product family design. While the interviewees in the interview study in chapter 3 have doubts for the applicability of product family design, they assign the model database with modular models a high level of applicability (see section 3.2.3). Thus, the approach uses the model database to enable product family design.

Divide et impera.

Divide and rule is the main idea of the model database. It is a contribution for handling complexity. The system behaviour of most products is highly complex (see chapter 3). In terms of ADGs this means there are many attributes and dependencies. The whole system is cut into smaller pieces: modular models. A modular model is a standalone model with defined interfaces which can be

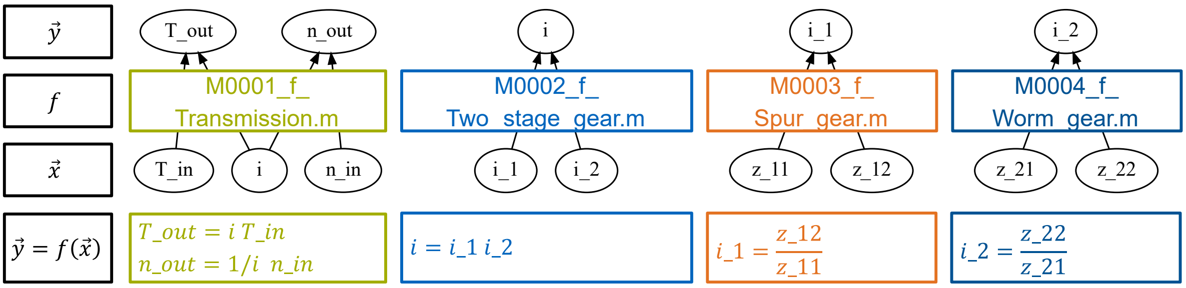


Figure 6.6: Modular models of the simple transmission example. Figure according to Rötzer et al. (2020a)

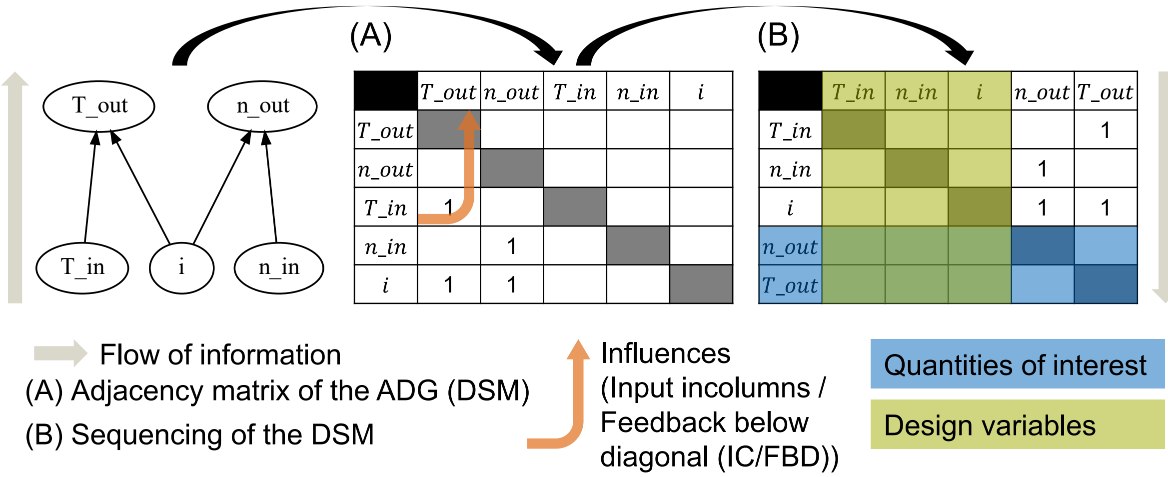


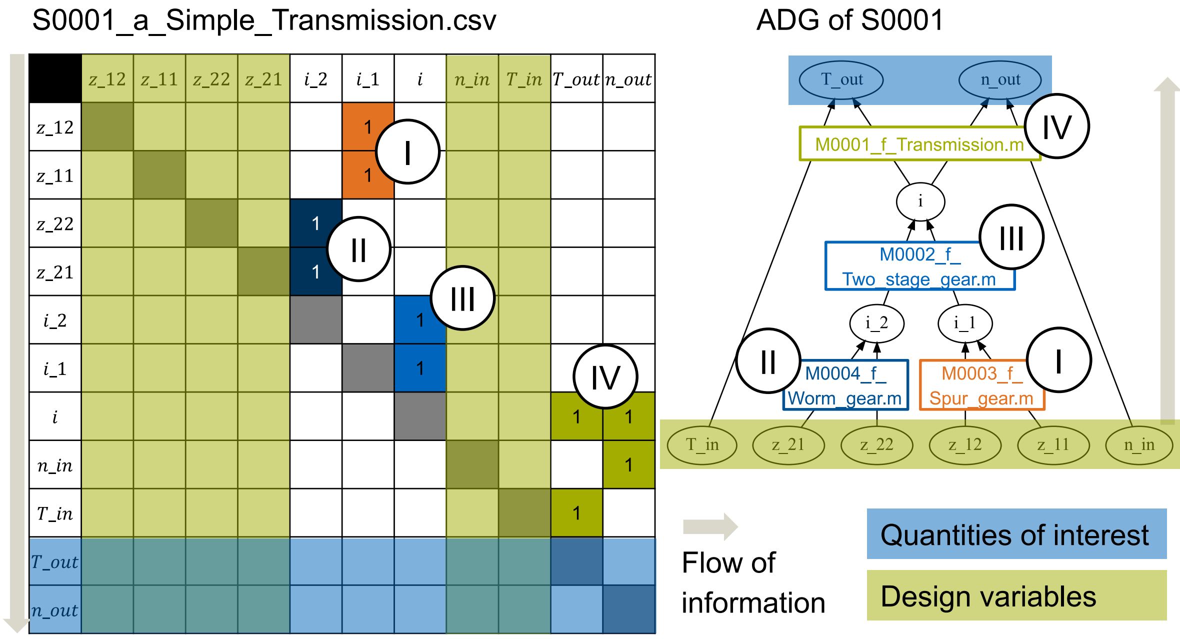
Figure 6.7: (A) Converting an attribute dependency graph into a design structure matrix. (B) Sequencing of the design structure matrix and identification of design variables and quantities of interest. Figure according to Rötzer et al. (2020a).

(re-)used in different contexts. Those modular models are built and validated. Then an algorithm merges them automatically to the system model.

Figure 6.6 shows the quantitative models of the simple transmission example. Each of the four models has a unique ID, starting with a 'M' for modular model. They quantify parts of the transmission and have their own ADGs with with quantities of interests and design variables. In this case, the bottom-up mappings are simple formulas. In general, bottom-up mappings need to quantify the dependencies from a lower to a higher hierarchical level of an ADG.

An algorithm merges those modular models to a system model. It uses the information of the ADG, depicted as a design structure matrix (DSM) and sequencing. Figure 6.7 shows this process. An ADG can be converted into its adjacency matrix which is a DSM. The attributes are the rows and columns of a quadratic matrix. The dependencies are entered as '1's according to the input in columns/ feedback below diagonal (IC/FBD) convention. Thus influence of attributes on other attributes can be read from the rows to the columns. In the example T_{in} influences T_{out}. When an attribute has no entries in its column it means that it is not influenced by other attributes. It is a design variable. When an attribute does not have any entry in its row it means it does not influence any other attribute. It is a quantity of interest.

This notation is usually used in temporal DSMs to depict dependencies of activities within a process. Sequencing is used to define the order of execution of activities to avoid feedback or make feedback loops as small as possible. Due to the fact ADGs do not have circular dependencies, there will always be a solution without feedback, i.e., no entries below the diagonal. Thus, sequencing (see section 5.1) can always re-order the adjacency matrix (DSM) such that the models can be executed sequentially.




```
function [T_out,n_out] = S0001_f_SimpleTransmission (z_12,z_11,z_22,z_21,n_in,T_in)
         [i_1] = M0003_f_Spur_gear (z_11,z_12);
         [i_2] = M0004_f_Worm_gear (z_21,z_22);
         [i] = M0002_f_Two_stage_gear (i_1,i_2);
         [T_out,n_out] = M0001_f_Transmission (T_in,n_in,i);
end
```

Figure 6.8: Top: DSM of the merged system (S0001_a_Simple_Transmission.csv) and corresponding ADG. Bottom: Bottom-up mapping of the merged system (S0001_f_Simple_Transmission.m). Figure according to Rötzer et al. (2020a).

Figure 6.8 shows the merged system. It consists of the four modular models M0001 - M0004. The algorithm merges all DSMs of the sub models to an overall DSM and re-orders the attributes until there is a sequential order without loops. The order of function evaluation is determined by the order of appearance in the DSM (top-down): M0003, M0004, M0002, M0001. Roman numbers indicate the order of function evaluations. The DSM can be converted to an ADG again. Design variables and quantities of interest can change their role while being merged. i_1 was a QoI in M0003 and a DV in M0002. In the merged system S0001 it is an intermediate attribute. The algorithm identifies those attributes from different modular models automatically if they have the same name.

Table 6.4 illustrates how the models are stored in the database using the example. ADGs are stored as .csv files, bottom-up mappings as MatLab™file (.m). PowerPoint™ slides serve as fact sheets including all relevant information of the modular model and its ADG.

The calculation of solution spaces and optimization algorithms will refer to the bottom-up mapping of the system (here: S0001) to calculate the system response.

Modular model

	ID	M0001
	Name	Transmission
Files	ADG	M0001_a_Transmission.csv
	Bottom-up mapping	M0001_f_Transmission.m
	Fact sheet	M0001_p_Transmission.pptx

Merged system model

	ID	S0001
	Name	Simple_Transmission
Files	ADG	S0001_a_Simple_Transmission.csv
	Bottom-up mapping	S0001_f_Simple_Transmission.m

Table 6.4: Structure of the model database. Top: names and files of a modular model. Bottom: name and files of a merged system model.

6.6 Step 4: Model Cost

"Cost considerations have been the main concern in regard with family design and platform development [...]" (Simpson et al., 2014, p. 24). As the costs are the main driver for product family design, this approach uses a cost model as an objective function of the optimization algorithm. Experience from an consulting company specialized on variant management revealed that companies search for a cost-optimized degree of standardization. Therefore, a cost model needs to incorporate relevant effects of standardization on costs. Relevant effects are:

- design related effects, such as the influence of design variable values on manufacturing costs or purchase costs.
- effects related to the handling of internal variety, such as storage, tooling or development of component variants.
- effects of economies of scale, such as discounts or learning effects during production for high volumes.

Figure 6.9 illustrates the proposed cost model as an attribute dependency graph. It provides an overview. The detailed description follows in equations (6.1) - (6.3). The overall cost of the product family is C_{PF}. It consists of design related costs $C_{\mathbf{x}}$ and costs which arise from creating component variants independently from their design $C_{n_{CV}}$.

$C_{\mathbf{x}}$ depends on the following attributes:

- $\hat{c}_{x_j}$: function for design related costs of component j, such as manufacturing or purchase costs
- $\hat{d}_j$: function for economies of scale, such as discounts
- $N_{j,i}$: production volume of component variant i of component j
- $\mathbf{x}$: design variables of the component variants

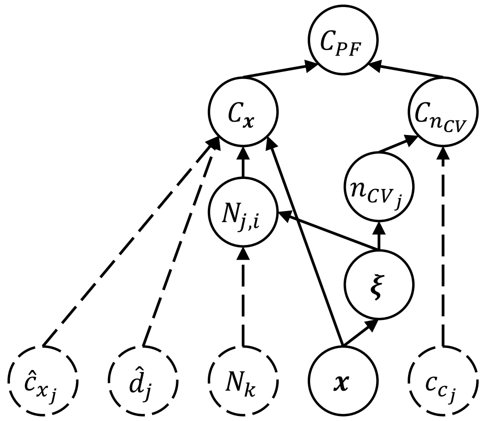


Figure 6.9: ADG of the proposed cost model

	Product $k=1$ $N_{k=1} = 10,000$	Product $k=2$ $N_{k=2} = 12,000$	Product $k=3$ $N_{k=3} = 8,000$	n_{CV_j}
Component $j=1$	$N_{j=1,i=1} = 22,000$		$N_{j=1,i=2} = 8,000$	$n_{CV_{j=1}} = 2$
Component $j=2$	$N_{j=2,i=1} = 10,000$	$N_{2,2} = 12,000$	$N_{2,3} = 8,000$	$n_{CV_2} = 3$
Component $j=3$	$N_{3,1} = 30,000$			$n_{CV_3} = 1$

Table 6.5: Example of production volumes of component variants according to the production volumes of products and the assignment scheme. The number of component variants is n_{CV}.

- ξ: assignment scheme

x determines the assignment scheme ξ. Component variants are shared between product variants when the design variables have the same value. The assignment scheme ξ and the production volumes of the different products k N_k determine the production volumes of the component variants. Table 6.5 illustrates this dependency with an example. If a component variant i is shared between different product variants k, the production volumes of the components will be summed up. This value is the production volume of the component variant i. This is done for all components j.

$C_{n_{CV}}$ depends on the following attributes:

- n_{CV_j}: number of component variants of component j
- c_{c_j}: complexity costs per component variant of component j

The assignment scheme ξ determines the number of component variants n_{CV_j}. Table 6.5 shows the number of component variants n_{CV_j} for each component and their production volumes $N_{i,j}$. Equations (6.1) - (6.3) calculate the overall cost of the product family.

$$C_{PF} = C_{\mathbf{x}} + C_{n_{CV}} \tag{6.1}$$

$$C_{\mathbf{x}} = \sum_{j}^{n_C} \sum_{i}^{n_{CV}} N_{j,i} \; \hat{c}_j(x_{j,i}) \; \hat{d}_j(N_{j,i}) \tag{6.2}$$

$$C_{n_{CV}} = \sum_{j}^{n_C} n_{CV_j}\, c_{c_j} \tag{6.3}$$

$j = 1,...,n_C$	component index	n_C	number of components
$i = 1,...,n_{CV_j}$	component variant index	n_{CV_j}	number of component variants of component j
$k = 1,...,n_P$	product variant index	n_P	number of product variants
$N_{j,i}$	production volume of component variant i of component j		
$\hat{c}_j(x_{j,i})$	design-dependent cost function of component j		
$\hat{d}_j(N_{j,i})$	function of effects of economies of scale for component j		
c_{c_j}	complexity costs per component variant of component j		
$x_{j,i}$	design variable values for component j component variant i		
$\xi_{k,j} \in 1,...,n_{CV_j}$	assignment scheme that assigns each product variant k and component j one component variant		
$\zeta_j^{\alpha \leftrightarrow \alpha+1} \in 0,1$	commonality pattern that indicates whether component j is shared between product variants α and $\alpha+1$; $\alpha = 1,...,n_P - 1$		

The task during the application of the approach is to determine all attributes which are indicated with dashed lines in Figure 6.9. $\hat{c}_j(x_{j,i})$ and $\hat{d}_j(N_{j,i})$ are functions. $\hat{c}_j(x_{j,i})$ can be determined by analyzing prices of components when being purchased. Manufacturing costs can be modeled along typical scaling variables, such as material volume, surface or other dimensions. Complexity costs arise by introducing new component variants. They represent costs which result from internal complexity and handling variety, such as development costs or storage. They may have many different sources and are thus hard to quantify. Some companies have costs per item number which can be used here. Ehrlenspiel et al. (2020) give an overview of cost calculations and estimations.

As those costs are mostly hard to quantify precisely, the user can define scenarios for costs and production volumes. The user can define intervals for each uncertain parameter. This increases the chance that the true value is in between the limits. Those scenarios can also be used to estimate future trends (Lindemann, 2016).

For the simple transmission example costs and production volumes are modeled as follows.

$j = 1,2,3,4$ is the component index. In this case, each component is described by one design variable[1], the number of cog teeth z_j. In this example the design-dependent cost function is linear with a constant scaling factor c_j and a constant for fixed costs c_{fix_j} (see equation (6.4).

$$\hat{c}_j(x_{j,i}) = c_j\, x_{j,i} + c_{fix_j} \tag{6.4}$$

Equation (6.5) shows the function to calculate the effects of economies of scale $\hat{d}_j$ for the simple transmission example. d_j is the discount on the design-related costs on component j when the production volume doubles. $min(N_k)$ is the minimum production volume among all considered

[1]The approach can also handle components with more than one DV.

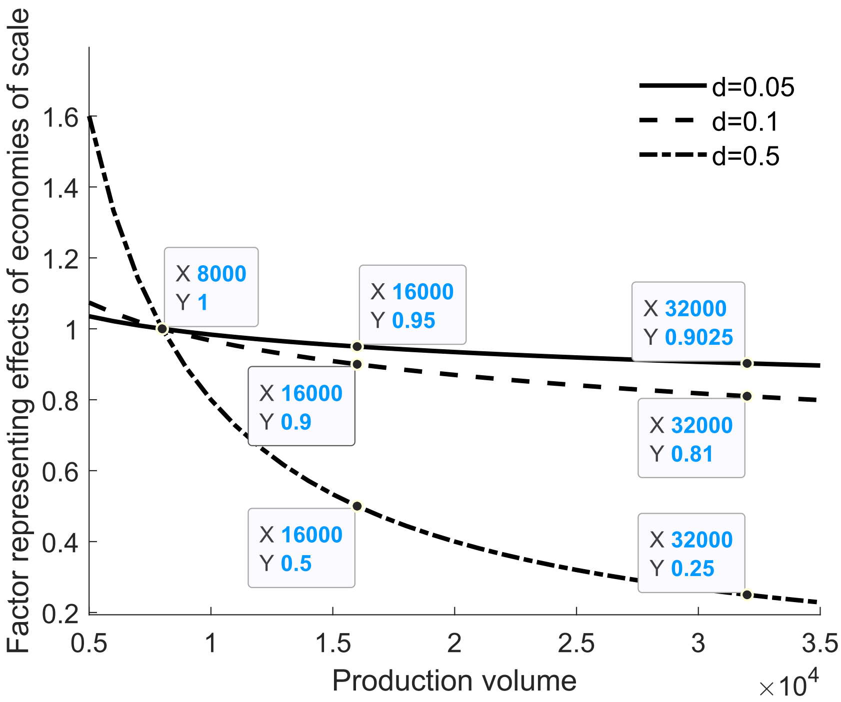


Figure 6.10: Effects of economies of scale $\hat{d}_j$ for different values of d_j according to equation (6.5)

Component j	1	2	3	4	unit
x_j	$z_{1,1}$	$z_{1,2}$	$z_{2,1}$	$z_{2,2}$	–
c_j	0.5	0.5	2.5	1.5	€/number of cog teeth
c_{fixed_j}	7.5	7.5	15	10	€
d_j	0.05	0.05	0.05	0.05	–
c_{c_j}	3000	3000	3000	3000	€/component variant

Table 6.6: Parameters of the cost model of the simple transmission example

product variants k. It serves as the reference production volume. Table 6.6 shows values for d_j used for this example.

$$\hat{d}_j(N_{j,i}) = \frac{(1-d_j)^{log_2(N_{j,i})}}{(1-d_j)^{log_2(min(N_k))}} \tag{6.5}$$

Figure 6.10 illustrates the effects of economies of scale for different values of d_j according to equation (6.5). Table 6.5 provides the production volumes used for this figure. $Min(N_k) = 8000$ is the reference value. Thus, for all curves there is no effect of economies of scale ($\hat{d}_j = 1$). When production volumes double the discount d_j applies on $\hat{d}_j$. For $d = 0.5$, $\hat{d}_{d=0.5}(N_{j,i} = 16{,}000) = 0.5$. When doubling the production volume again $\hat{d}_{d=0.5}(N_{j,i} = 32{,}000) = 0.25$.

The approach proposes this cost model as an objective function. It is used to identify the cost-optimized assignment scheme and design variable values of the components. It does not need to quantify the actual costs exactly but needs to incorporate relevant effects. Other objective functions of a similar structure can be implemented without additional effort.

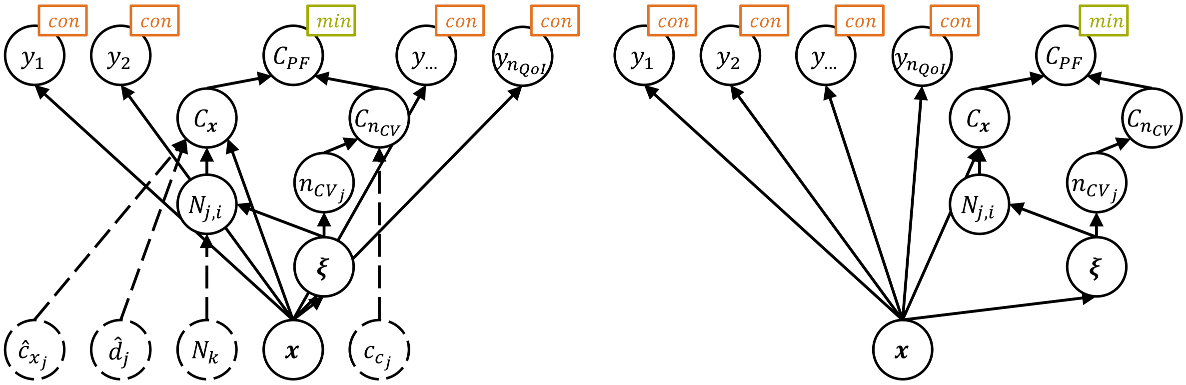


Figure 6.11: ADG to describe the design problem. Left: ADG with design parameters of cost model. Right: re-arranged ADG containing only design variables. con: constrained; min: minimized.

6.7 Step 5: Optimize Product Family

This step uses all the information collected in the steps before to optimize the product family.

Figure 6.11 shows the ADG for the optimization problem. On the left the ADG from the cost model (see figure 6.9 in section 6.6) is extended by the quantities of interest from the technical model $y_1, \dots, y_{n_{QoI}}$ (see section 6.4). The requirements from section 6.3 constrain those quantities of interest. The overall product family cost C_{PF} needs to be minimized. Figure 6.11 right shows the re-arranged ADG without the design parameters from the cost model.

Equations 6.6 show the mathematical problem statement of the corresponding product family cost optimization problem.

$$\begin{aligned} \min_{\mathbf{x}} \quad & C_{PF}(\mathbf{x}) \\ \textit{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u \end{aligned} \tag{6.6}$$

$C_{PF}(\mathbf{x})$ is the objective function defined by equations (6.1) - (6.3) in section 6.6. $\mathbf{g(x)}$ and $\mathbf{h(x)}$ are the constraints which ensure the fulfillment of the requirements on the quantities of interest $\mathbf{y}$ from section 6.3. $\mathbf{x}^l$ and $\mathbf{x}^u$ define the lower and upper boundaries of the design space respectively.

To solve this optimization problem, the user can choose between two approaches: (I) cost-optimized product family design using solution space engineering (PFD-SSE) or (II) an optimization algorithm scanning the complete design space (PFD-CDS).

Cost-optimized product family design using solution spaces (PFD-SSE)

PFD-SSE (I) is a two-level approach. On the first level it uses solution spaces to define permissible intervals for the design variable values. This is based on the approach by Eichstetter et al. (2015) (see section 5.3). On the second level the cost of the product family is optimized within these intervals.

Figure 6.12 shows the solution spaces for the three product variants of the simple transmission problem. The four design variables (see figure 6.3, left) constitute a four-dimensional design space. The diagrams in figure 6.12 are two-dimensional projections of that four-dimensional space. One

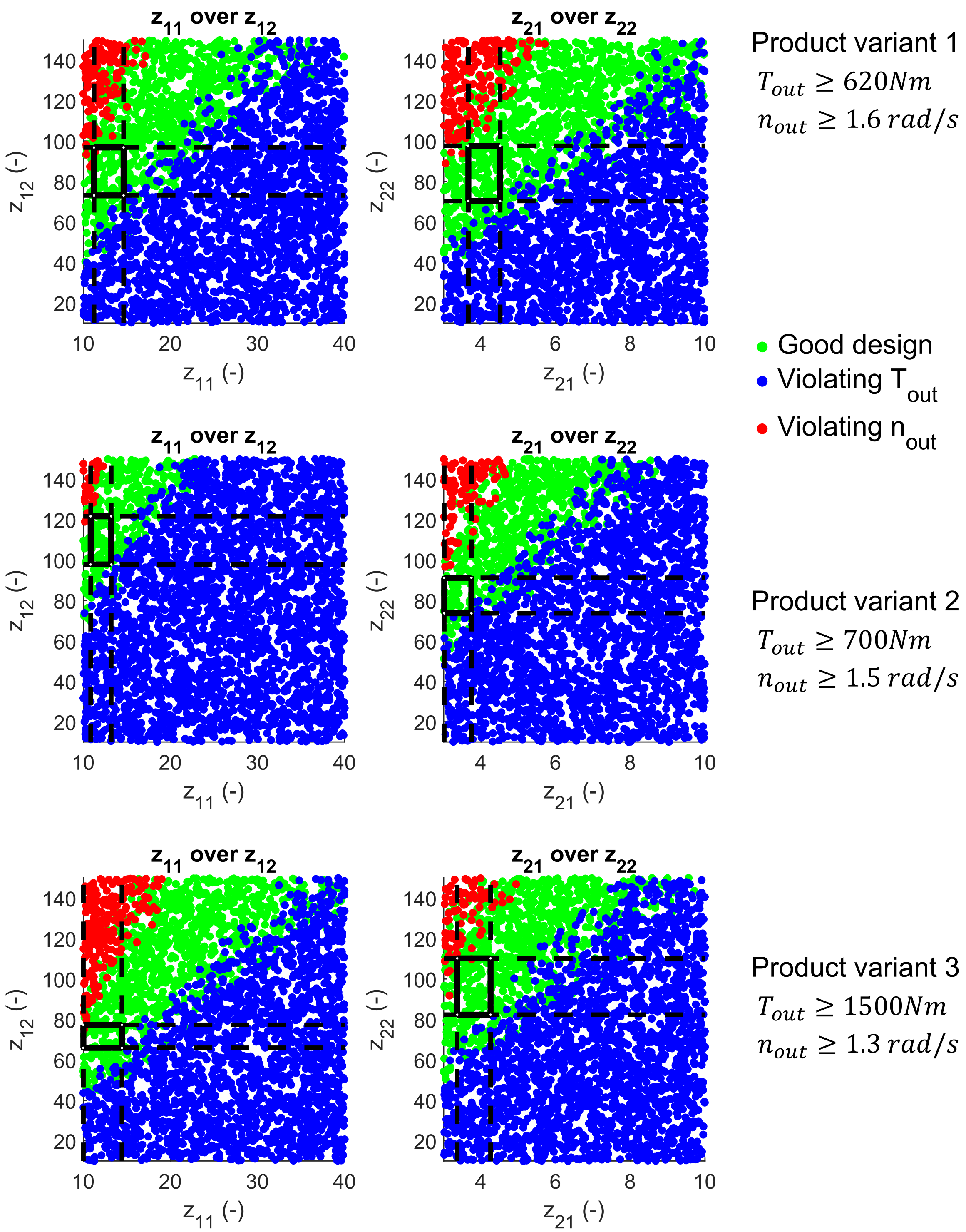


Figure 6.12: Solution Spaces for the three product variants of the simple transmission problem

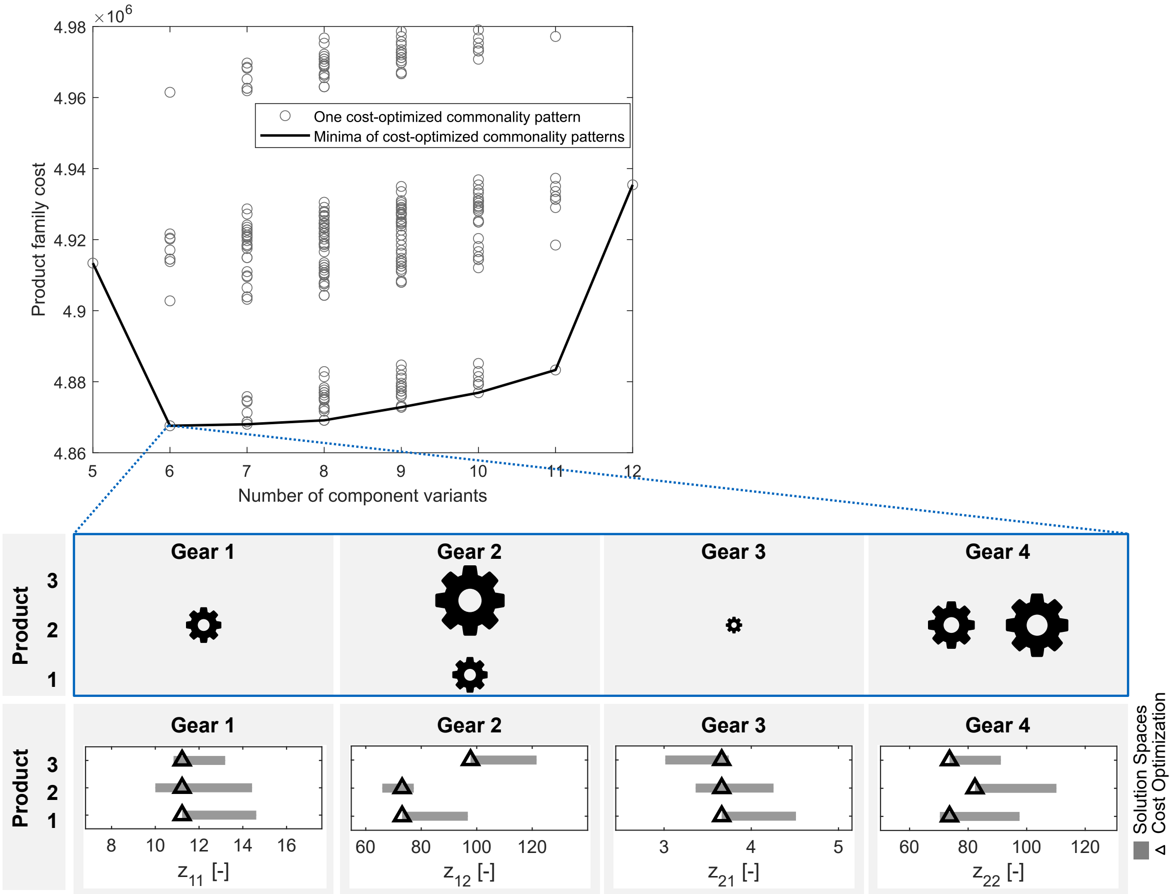


Figure 6.13: Optimized assignment scheme and design variable values for a product family of transmissions using PFD-SSE. Top: total cost of the product family over number of components variants. Middle: optimized assignment scheme with components. Bottom: Optimized design variable values of the component variants. Intervals derived from the box-shaped solution spaces (see figure 6.12). Triangles represent optimized design variable values.

point represents one possible design generated using random sampling. The merged system model from figure 6.8 calculates the system response of each point. Then this system response is compared to the requirements from table 6.3. If a point fulfills all requirements, it is green. If it violates at least one it is drawn in the colour of the respective requirement it is violating. Here blue points violate the required output torque T_{out}. Red points violate the required output rotational speed n_{out}. The green dots fulfill all requirements and constitute the solution space. The black rectangle represents the box-shaped solution space. The diagrams on the left represent the cog teeth of the first transmission stage (spur gear). The diagrams on the right shows the cog teeth of the second transmission stage (worm gear). The values for the requirements differ for the three product variants. Thus, the solution spaces look different for each product variant. The user defines box-shaped solution spaces for each product variant. These box-shaped solution spaces include only good designs and decouple the design variables from each other. As long as each design variable stays within its intervals, the other design variables can assume arbitrary values **within** their intervals. When solution spaces of different product variants overlap, it means that the design variables fulfill the requirements of two or more product variants. This means that the components which correspond to the design variables can be made the same.

Among all possible degrees of commonality resulting from technical analysis with solution spaces, the algorithm searches for the cost-optimized one. Figure 6.13 shows the results of the cost optimization. The diagram on the top shows the overall product family cost C_{PF} with respect to the number of component variants used in the product family. The number of component variants in this figure is the total number of all component variants within the product family and does not distinguish between the different components. Thus, six component variants can be realized with one variant of gear 1, one variant of gear 2, two variants of gear 3 and two variants of gear 4, as well as, with 3 variants of gear 1, one variant of gear 2, one variant of gear 3 and one variant of gear 4. All these possibilities are indicated by dots.

Five component variants is the highest degree of standardization possible. Twelve components variants result from individual component variants for each of the three components of the product family consisting of four product variants ($3 \cdot 4 = 12$). In between all subsets of commonality are possible. According to Bell's number, there are $(B_3)^4 = 625$ possibilities to share component variants in a product family with three product variants and four components. Each dot in the diagram represents one possibility which is possible from a technical perspective. These dots represent cost-optimized product families for a specific assignment scheme. The black line connects the local minima of different number of component variants. The minimum of this line represents the cost-optimized product family design.

Here it consists of six component variants. Using five component variants would lead to over-dimensioning of the component variants resulting in higher costs. Using more than six component variants also increases the costs. This is due to decreasing effects of economy of scales and increasing complexity costs. This minimum is highly dependent on the application example and its cost function. The design variable values and the assignment scheme of the cost-optimized product family are also output of the algorithm. They are shown in the middle and at the bottom of figure 6.13. The box-shaped solution spaces provide directly the intervals of the design variables for each product variant. Overlapping intervals indicate potential for standardization. The optimization algorithm identifies the cost-optimized assignment scheme (black rectangle) and the cost-optimized design variable values for each component variant (triangle). This optimization took 16s on an Intel® Core™i7-1065G7 CPU @ 1.30GHz.

Figure 6.14 abstracts the procedure of product family design using solution spaces. At first, the user determines box-shaped solution spaces for all product variants. Then the algorithm identifies overlaps between the different box-shaped solution spaces. In the last step all possible combinations are optimized. This leads to the results shown in figure 6.13.

Figure 6.15 shows the algorithm used for the cost optimization. The algorithm calculates all possible assignment schemes according to Bell's number (Bell, 1934). Then the algorithm searches for the cost-optimized assignment scheme (exhaustive search).

In both cases the algorithm at first checks whether a specific assignment scheme is feasible from a technical perspective, i.e., the solution spaces are overlapping. If this applies, the common interval of design variable values for the given assignment scheme is determined. Within the limits of the overlapping solution space, a gradient-based non linear optimization algorithm determines the cost-optimized design variable values (see section 5.4). This optimization does not need to calculate the system response to check the constraints as they are already ensured by the box-shaped solution spaces.

Figure 6.16 uses ADGs to illustrate how cost-optimized product family design using solution spaces (PFD-SSE) solve the general problem statement from figure 6.11. It combines a technical perspective (A) with an economical perspective (B). During the solution space analysis (A) only the technical system is analyzed. The result, box-shaped solution spaces for each product variant and design variable, determine then the potential for standardization from a technical perspective, including the

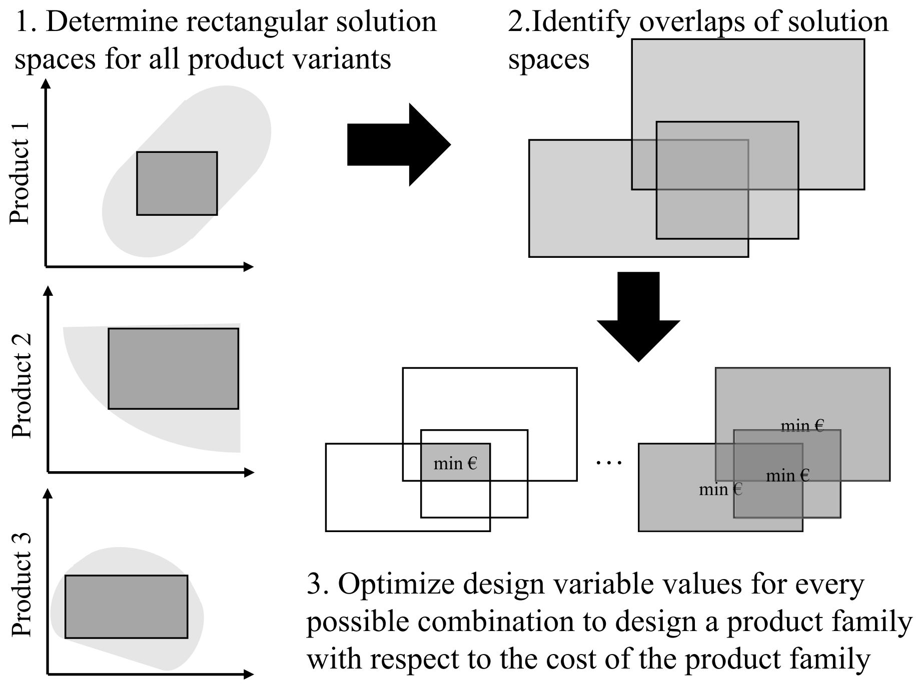


Figure 6.14: Procedure model of product family design using solution spaces according to (Rötzer et al., 2020b).

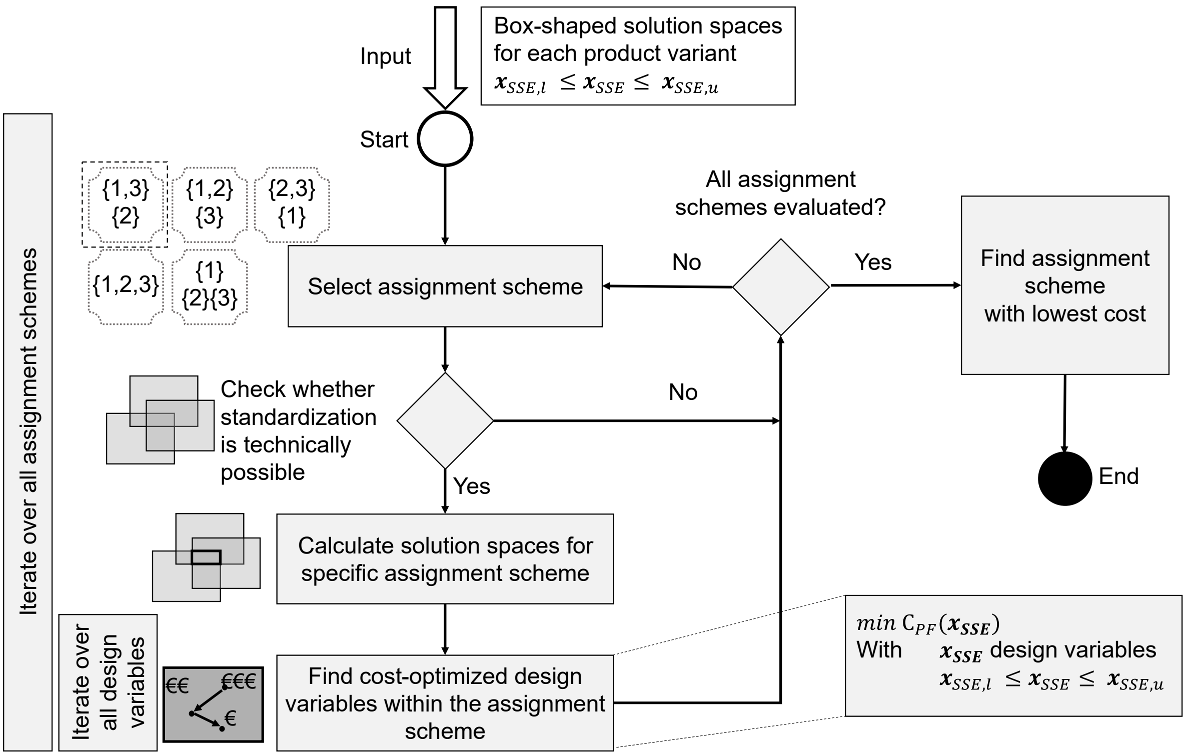


Figure 6.15: Algorithm to optimize the product family based on solution spaces (steps 2 and 3 in figure 6.14). Figure adapted from Rötzer et al. (2020b).

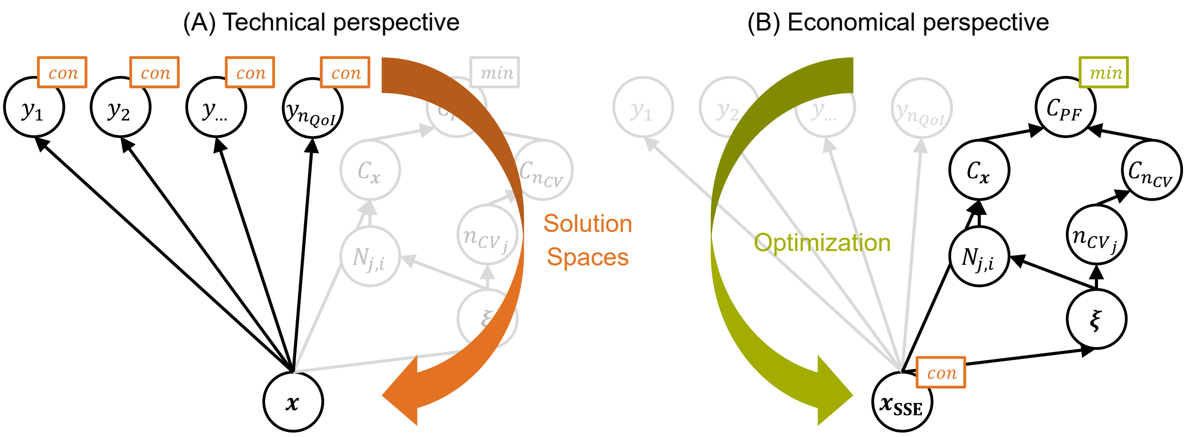


Figure 6.16: Cost-optimized product family design using solution spaces (PFD-SSE) described with ADGs.

DV	Unit	Product 1	Product 2	Product 3
$z_{1,1}$	-	10		
$z_{1,2}$	-	88		146
$z_{2,1}$	-	3		
$z_{2,2}$	-	34		
Product family cost	€	3,230,696		

Table 6.7: Cost-optimized assignment scheme and design variable values of the simple transmission problem using PFD-CDS

fulfillment of all requirements on the product family. In (B) all possibilities for standardization from a technical perspective are optimized according to the overall cost of the product family. Here, the technical system is not considered explicitly but implicitly by constraining the design space of x to the box-shaped solution space. This allows for a fast optimization of different assignment schemes.

Cost optimization of product families using the complete design space (PFD-CDS)

PFD-SSE is limited because not the entire design space is considered. The goal now is to formulate a general approach that considers the entire design space. It must be applicable to various problems statements without constraints, such as linearity or continuity. It must be built from the algorithms of existing approaches (see section 2.3) and, thus, serve as a benchmark.

This algorithm shall provide a point-based solution which fulfills the requirements on the quantities of interest and minimizes product family cost by varying both the commonality pattern and the design variable values.

PFD-CDS also uses the requirements from section 6.3 to constrain the quantities of interest. It uses the system model from section 6.5 to calculate the system response. The cost model from section 6.6 serves as an objective function.

Table 6.7 shows the results using the PFD-CDS algorithm. The calculation took 336s on an Intel® Core™i7-1065G7 CPU @ 1.30GHz. This is a point-based solution without information about the solution spaces. Here, the cost-optimized assignment scheme consists of five product variants with one variant for gear 1,3 and 4; and two variants for gear 2. The overall costs are lower than using PFD-SSE (see 6.13). PFD-SSE uses only the design variable values within the box-shaped solution spaces for the optimization process. Thus, all solutions which are not in this box-shaped solution space are omitted. In this simple example PFD-CDS can scan the complete design space and find a solution with lower cost. If the box-shaped solution spaces include the optimal design variable values from PFD-CDS it will also find the same optimum.

Equation (6.7) shows the problem statement of the optimization problem being solved by PFD-CDS. It is a nested optimization problem with two levels: (1) the commonality pattern is optimized and (2) the components are optimized for each commonality pattern from level (1). **g(x)** and **h(x)** are problem specific constraint functions.

$$
\begin{aligned}
&(1) && \min_{\zeta \in Z} \overline{C}_{PF}(\zeta) \\
&(2) && \overline{C}_{PF}(\zeta) = \min_{\mathbf{x}} C_{PF}(\mathbf{x}, \zeta) \\
&\textit{subject to} && \mathbf{g}(\mathbf{x}) \leq 0 \\
& && \mathbf{h}(\mathbf{x}) = 0 \\
& && \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u \\
&\textit{with} && Z = \{\zeta_j^{\alpha \leftrightarrow \alpha+1} | \alpha = 1,...,n_P - 1;\ j = 1,...,n_C\}
\end{aligned}
\tag{6.7}
$$

Equation (6.8) shows the definition of the commonality pattern used by PFD-CDS algorithm. This commonality pattern is similar to the so-called commonality and similarity pattern of Fujita and Yoshida (2004). However, there are two limitations: First, the commonality pattern in equation (6.8) only allows for commonality and not for similarity, i.e., shared component variants need to be the same with respect to all design variables that describe the component. Second, the commonality pattern in equation (6.8) only allows for commonality between neighbouring product variants α and $\alpha + 1$.

$$
\zeta_j^{\alpha \leftrightarrow \alpha+1} = \begin{cases} 1, \text{ if component variant is shared between product variants } \alpha \text{ and } \alpha + 1 \\ 0, \text{ if component variant is not shared between product variants } \alpha \text{ and } \alpha + 1 \end{cases}
\tag{6.8}
$$

$$
\begin{aligned}
\textit{with}: & \ j = 1,...,n_C \\
& \ \alpha = 1,...,n_P - 1
\end{aligned}
$$

Figure 6.17 illustrates how PFD-CDS solves the product family design problem. In contrast to PFD-SSE, the system response needs to be calculated during the optimization to check whether the constraints are fulfilled.

PFD-CDS is a coupled two-level approach. Figure 6.18 illustrates the procedure of the algorithm: on the upper level it varies the commonality pattern. On the lower level it optimizes the design variable values for a specific commonality pattern from the upper level. For each optimization step the system response function needs to be evaluated to check whether the constraints on the quantities of interest are fulfilled.

The upper level uses a genetic algorithm (GA). GAs are well suited to solve integer problems (see chapter 2 and 5). The lower level uses a particle swarm optimization (PSO) to identify the best design of the commonality pattern from the upper level (see section 5.4). Chowdhury et al. (2011) claims that PSO is broadly applicable in product family design. The application of PSO is not constrained to

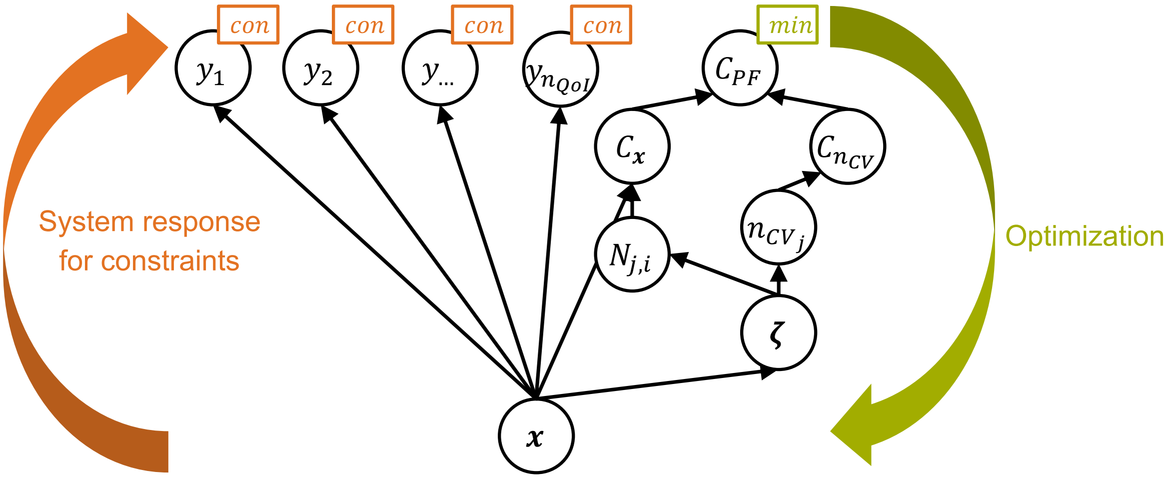


Figure 6.17: Cost-optimized product family design using the complete design space (PFD-CDS) described with an ADG.

continuous functions or variables.

Figure 6.18 illustrates the two-level algorithm to optimize product families using the complete design space (PFD-CDS). It incorporates the following two levels: the commonality pattern on the upper level and the design variable values on a lower level. A genetic algorithm optimizes the commonality patterns on the upper level; a particle swarm optimization algorithm the design variable values on a lower level. The lower level includes a routine which avoids multiple optimizations of yet optimized commonality patterns. Both optimization heuristics GA and PSO are stochastic. This means, especially for complex problems, the results may vary when being applied several times.

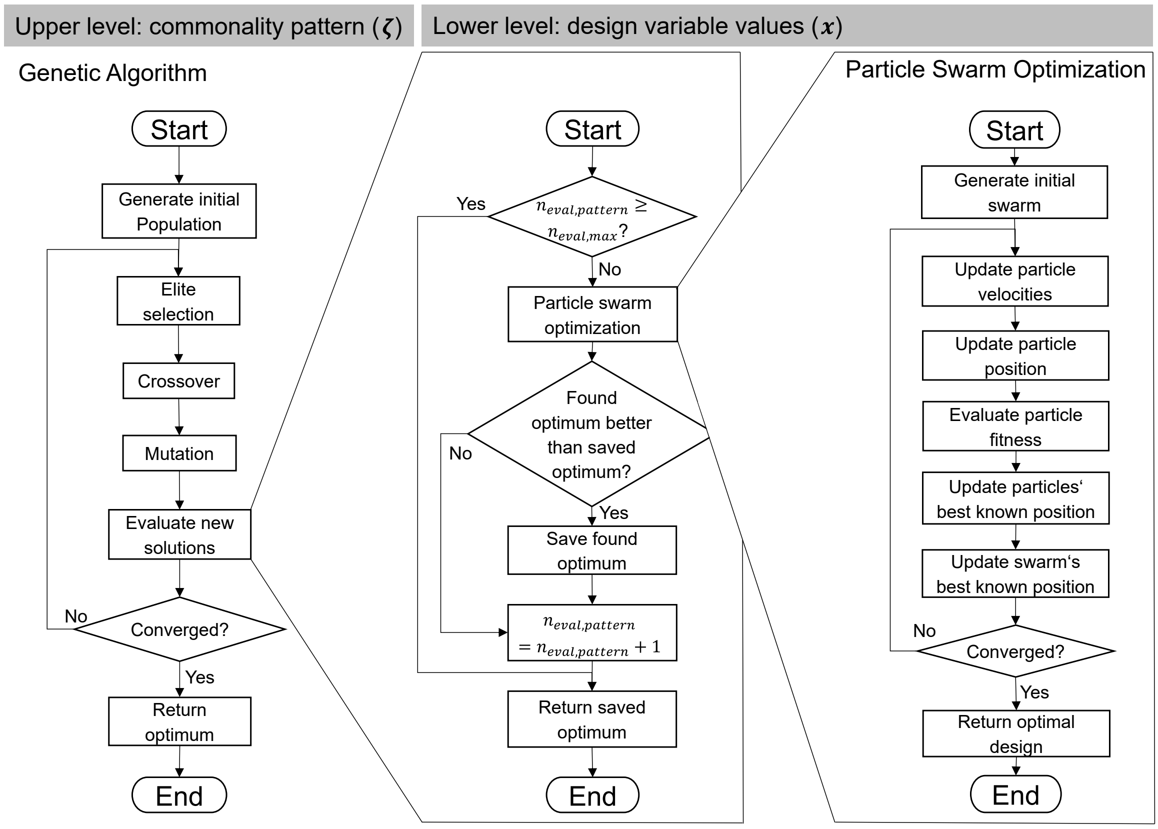


Figure 6.18: Algorithm used for cost-optimized product family design using the complete design space according to Rötzer et al. (2021).

7 Evaluation

The developed approach is evaluated with two application examples: section 7.1 shows the application to a product family of electric vehicles; section 7.2 to a product family of water hose boxes.

7.1 Evaluation Case 1: Electric Vehicle Design

This evaluation case describes the results of a cooperation with a startup company. Parts of the results presented here have been published in Rötzer et al. (2022a).

7.1.1 Introduction

The startup in this evaluation case fosters the development of a new architecture for electric vehicles. The design of purely electric vehicles enables new design possibilities. Figure 7.1 shows the rolling chassis developed by the startup. The basic idea is a flat chassis to which different structures can be attached without constraining those structures. Companies with no or little experience in vehicle design can use this rolling chassis to design electric vehicles. Those companies do not have to consider the design of motor, battery or vehicle dynamics and can focus on new mobility concepts or user experience. This enables new players to enter the market.

The concept is based on an in-wheel motor with integrated inverter. It operates in areas of high efficiency which reduces the total cost of ownership. Furthermore, parts of the drivetrain, such as transmissions, bearings or differential gears, are no longer necessary. This further reduces the weight of the vehicle.

The goal for this evaluation case is to find a cost-optimized product family design for the following key components of the concept: battery, inverter and motor. This includes finding both the optimized assignment scheme and design variable values for these components. The startup wants to offer product variants for all common customer segments. The design of the components needs also to incorporate the use case scenarios in which the vehicle will be used. Smaller vehicles are rather used in urban use case scenarios. Larger vehicles are rather used for longer distances and motorway use

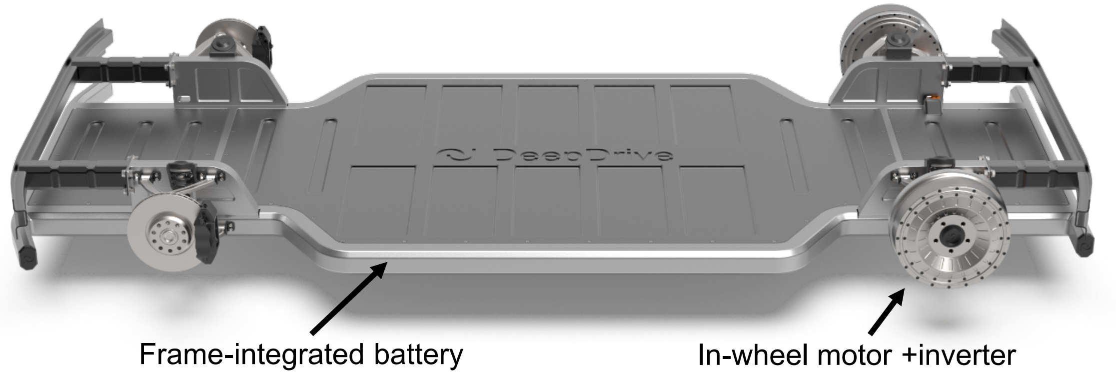


Figure 7.1: CAD rendering of rolling chassis including the frame-integrated battery and the in-wheel motors at the rear axle. Courtesy of DeepDrive GmbH.

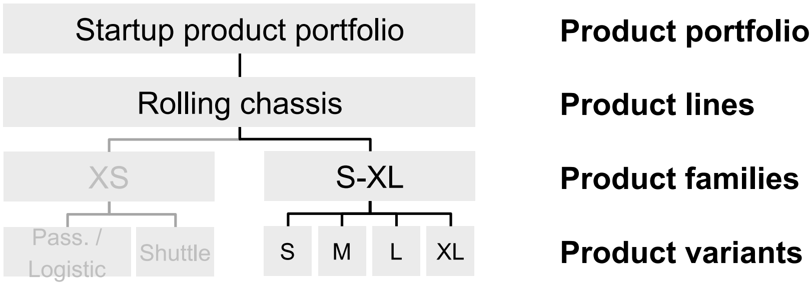


Figure 7.2: The startup's product portfolio with two product families.

case scenarios. The developed approach is applied during the early phase of product development. There is no information or experience available about predecessors. The product family design is subject to economical and technical uncertainty.

Figure 7.2 shows the startup's product portfolio consisting of a product line of rolling chassis and two product families: XS and S-XL. XS constitues an own product family. The homologation of the XS products differs from the other products. This leads to lower development, testing and complexity costs of this product family. Both product families were optimized during the project. This thesis presents the results of the S-XL product family.

Table 7.1 shows that the example meets the prerequisites of the developed approach.

ID	Prerequisite
PR 1	It is a scale-based product family. The components battery, inverter and motor are part of all product variants, but can be scaled differently.
PR 2	Requirements on the product variants are quantitative, such as range, velocity or acceleration.
PR 3	Product variants differ from each other by different requirements, e.g., range requirements.
PR 4	Design variables are quantifiable, i.e., the component variants differ from each other by the energy saved in the battery, values for current (inverter), and diameter and length (motor).
PR 5	System behaviour is quantifiable. There are driving cycles, engine maps and formulae available to describe the system behaviour of the vehicle.

Table 7.1: Prerequisites of the developed approach for evaluation case 1.

7.1.2 Step 1: Collect Requirements

The product variants should cover the market segments of common passenger and transport vehicles. A benchmark analysis was conducted to identify customer relevant requirements for the different product variants. Figure 7.3 shows the customer relevant requirements from the benchmark analysis. Ascent represents the climbing potential. Distance between motor and rim ensure that the motor fits the rim.

Product variant	Range	Acceleration 0-100 km/h	Ascent	Max velocity	Distance motor / rim
	[km]	[s]	[%]	[km/h]	[m]
S	≥500	≤7	≥40	≥150	≥0.03
M	≥600	≤7	≥40	≥150	≥0.03
L	≥600	≤12	≥30	≥150	≥0.03
XL	≥600	≤12	≥25	≥150	≥0.03

Figure 7.3: Quantities of interests and customer relevant requirements for the product variants S -XL of the electric vehicle (Rötzer et al., 2022a).

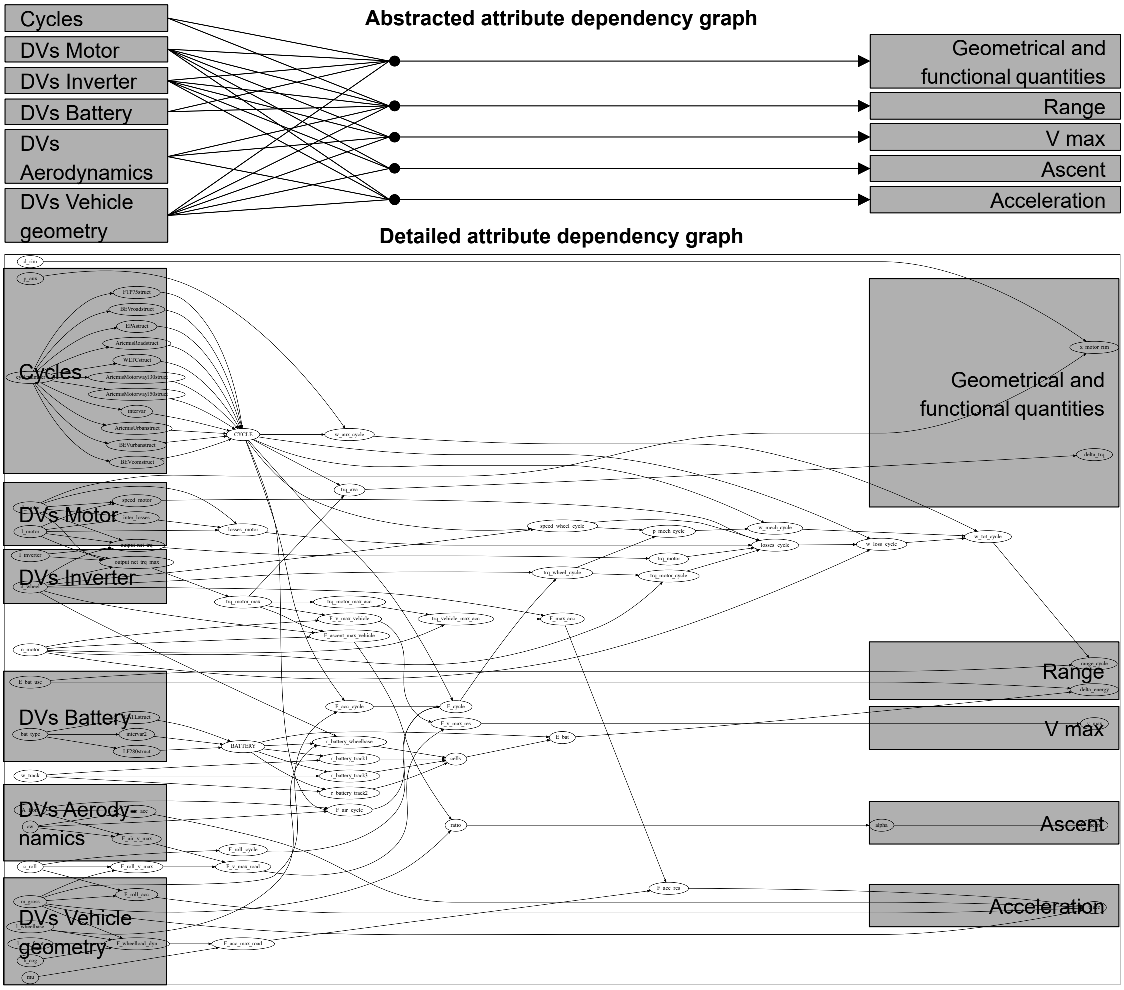
7.1.3 Step 2: Model System Dependencies

Figure 7.4 (top) gives an overview of the dependencies between design variables (DVs) on the left and quantities of interest (QoIs) on the right. The QoIs mostly match with the attributes in the requirement list (see figure 7.3). There are other requirements which are not stated in this list, such as that the electric vehicle is able to complete certain driven cycles, i.e., the motor provides enough torque at every point of the cycle. This is ensured by constraining the QoI δ_{trq}. This QoI is summarized in geometrical and functional requirements. The design attributes (left) are clustered in 6 categories: cycles, motor, inverter, battery, aerodynamics, and geometry. Although only battery, inverter, and motor need to be designed for the product family, the other attributes also influence the design. Thus, they need to be considered. For instance the vehicle geometry influences all groups of quantities of interest. The grouped attributes have the same dependencies as all its members. For example, the design variables of aerodynamics influence range, max velocity and acceleration, but do not influence ascent or geometrical requirements.

Figure 7.4 (bottom) allows for a in-depth analysis of the technical dependencies. It consists of 19 design attributes and seven QoIs. The following examples of dependency chains illustrate parts of the system dependencies.

As an example, we can follow the path of the design variables of the motor: diameter of the motor d_{motor} and length of the motor l_{motor}. d_{motor} has a dependency directly to $x_{motor,rim}$, which is the distance between motor and rim. d_{motor} and l_{motor} influence the engine maps, such as the speed-torque characteristic trq_{motor} or the speed-torque-loss characteristic $losses_{motor}$. Together with the torques and revolutions needed from the cycles, the latter determines the energy needed during the cycle $w_{tot,cycle}$. It consists of the mechanical energy $w_{mech,cycle}$ and the losses during the cycle $w_{loss,cycle}$. The design of the motor (d_{motor}, l_{motor}) influences the engine map. The engine map determines the available torque and the losses in a certain operating point of the motor during the cycle. The operating point determines the efficiency/losses of a specific motor. This again influences the energy needed during a cycle.

As pointed out in section 7.1.1, the design depends on the use case scenario described by cycles. This project takes ten cycles into account. The cycles influence the resistance forces, which a vehicle needs to overcome to drive the cycle: the air resistance force F_{air}, and the acceleration resistance force F_{acc}. Together with the rolling resistance F_{roll} and the diameter of the wheel, they influence the torque needed to complete the cycle. Together with the speed at the wheel, it influences the mechanical energy needed $w_{mech,cycle}$. Together with the losses of the drivetrain during the cycle $losses_{cycle}$ and the number of motors used the total energy needed for the cycle is determined $w_{tot,cycle}$. Together with the energy of the realized energy of the battery $E_{bat,use}$ the total energy needed determines the range.



DV	Description
d_{rim}	Rim diameter
p_{aux}	Power consumption of auxiliary components
$cyclenumber$	Selection of driving cycle
d_{motor}	Diameter of motor
l_{motor}	Length of motor
w_{track}	Track width
A_{front}	Vehicle front surface
c_w	Drag coefficient
c_{roll}	Rolling resistance coefficient
m_{gross}	Vehicle gross weight
$l_{wheelbase}$	Length of wheelbase
$l_{cog,front}$	Distance center of gravity to front axle
h_{cog}	Height of center of gravity
μ	Friction coefficient between wheel and surface
$I_{inverter}$	Inverter current
d_{wheel}	Wheel diameter
n_{motor}	Number of motors
$E_{bat,use}$	Energy content of battery in vehicle
bat_{type}	Selection of battery type used in vehicle

QoI	Description
$x_{motor,rim}$	Distance between motor and rim
δ_{trq}	Difference between available and necessary traction in cycle
$range_{cycle}$	Range according to driving cycle
δ_{energy}	Difference between available and necessary energy in cycle
v_{max}	Maximum velcocity
$ascent$	Climbing ability
t_{acc}	Acceleration time

Figure 7.4: Attribute dependency graph (ADG) of the electric vehicle. Left: Design variables and parameters. Right: Quantities of interest. Top: Abstracted ADG providing an overview of attributes and dependencies. Middle: Detailed dependency structure with intermediate attributes. Bottom: Description of DVs and QoIs (figure according to (Rötzer et al., 2022a)).

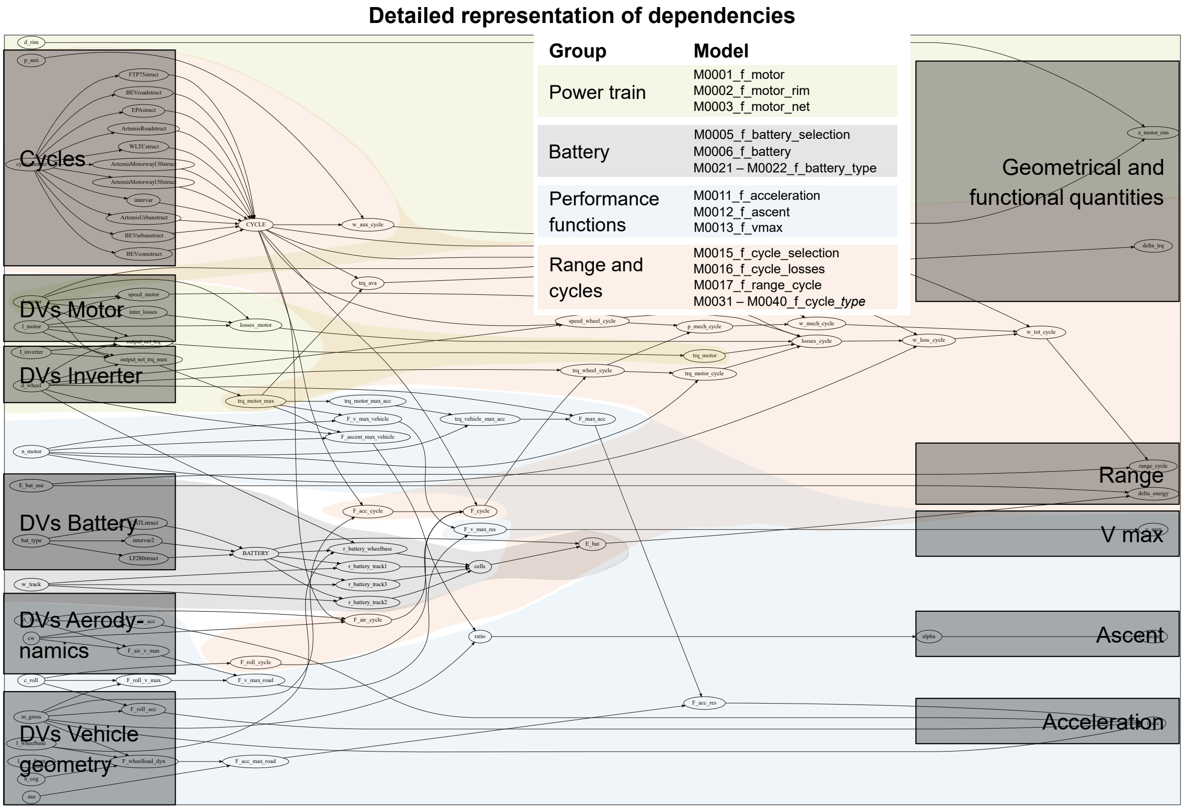


Figure 7.5: Modular models of the electric vehicle included in the attribute dependency graph (ADG) (Rötzer et al., 2020b).

Another example are the DVs of the vehicle geometry, such as the track width w_{track} and the wheelbase $l_{wheelbase}$. Both influence, together with the battery type, the number of cells ($cells$) included in the frame, which determine together with the voltage of the battery and the capacity of the cells the maximum possible energy content of the battery E_{bat}. At the same time $l_{wheelbase}$ influences the dynamic wheel load $F_{wheelload,dyn}$. $F_{wheelload,dyn}$ and the friction coefficient μ determine the max acceleration force which can be applied on the road $F_{acc,max,road}$. This can limit the max acceleration force from the motor $F_{max,acc}$, which depends on the design of the motor. Both $F_{max,acc}$ and $F_{acc,max,road}$ determine the resulting acceleration force $F_{acc,res}$, which is used to determine the QoI acceleration time t_{acc}.

The dependencies depicted in figure 7.4 incorporate knowledge from experts at the startup and literature (Mitschke and Wallentowitz, 2014).

7.1.4 Step 3: Model System Behaviour

In this step, the dependencies from the ADG in section 7.1.3 are quantified. Each arrow needs to be part of a model. Figure 7.5 shows all the modular models used to describe the system behaviour with their ID. In total the algorithm, presented in section 6.5, merges 23 modular models to create the system model.

To give a better overview, those 23 models are grouped into four categories: power train, battery, performance functions and range and cycles. Each group has a specific colour which highlights the models in the ADG. This assignment illustrates how to the different models interact.

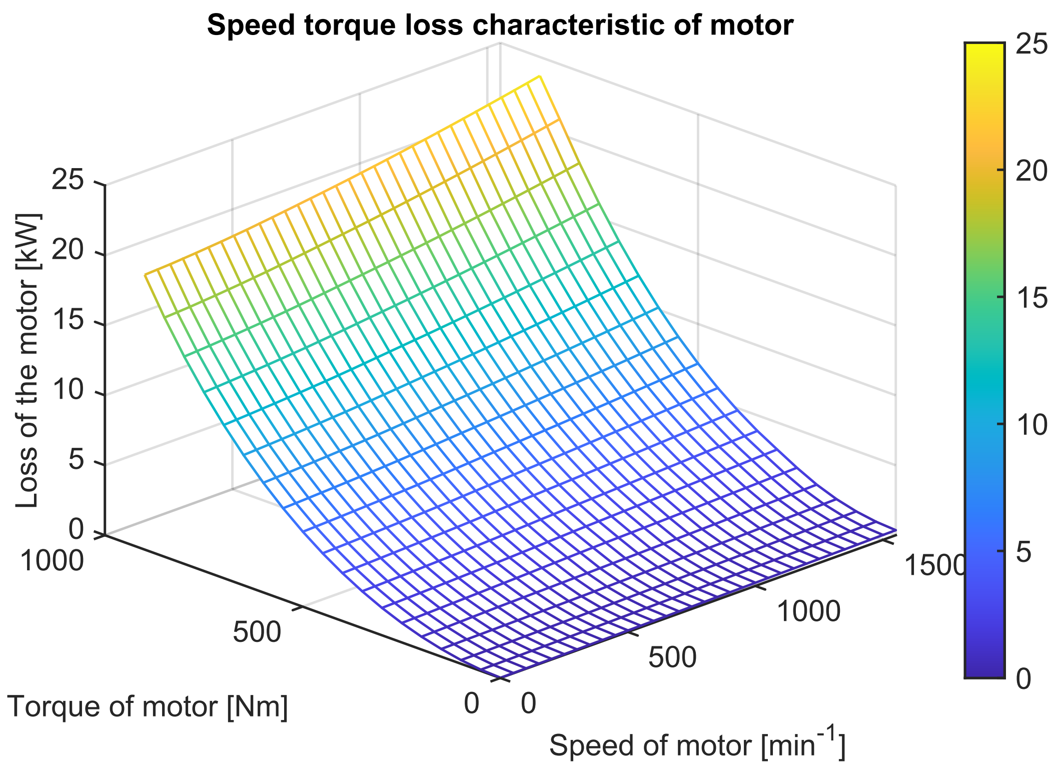


Figure 7.6: Speed-torque-loss characteristic ($losses_{motor}$) for one specific design of d_{motor}, l_{motor}, $I_{inverter}$.

The *power train* models quantify the interaction between motor and inverter to produce certain engine maps. *Battery* models offer the choice of two types of batteries. They calculate the number of cells which can be integrated in the frame and the resulting energy which can be stored. *Performance functions* calculate the QoIs acceleration, ascent and max. velocity according to established formulae from literature (Mitschke and Wallentowitz, 2014). *Range and cycle* models offer ten different cycles to choose from. They can be also evaluated simultaneously for different designs. Then, they calculate the range according to the selected cycles.

The following modular models represent parts of the system model.

M0003 provides design-specific speed-torque, and speed-torque-loss characteristics. The calculations are based on simulations of the experts at the startup. The results of the simulation are validated in tests. To reduce calculation time, neural networks are trained to provide the characteristics in shorter calculation times than the simulation.

Figure 7.6 shows the speed-torque-loss characteristic for one specific motor and inverter design, i.e., specified values for d_{motor}, l_{motor} and $I_{inverter}$. The three attributes describe the motor characteristic: the torque of the motor, the speed of the motor, and the loss of the motor. The characteristic strongly depends on the motor and inverter design. The characteristic depicted in figure 7.6 shows that the loss of the motor increases non-linearly with the torque. For higher torques, increasing speed increases the loss non-linearly. It can be seen that the engine map has a strong influence on the energy needed.

M0015 selects the driving cycle which is used to calculate the required torques and speeds of the motor. According to the design variable values of *cycle* the algorithm chooses the corresponding driving cycle.

Figure 7.7 gives an overview of the available cycles. They are grouped into three categories according

	Urban		Road				Motorway			
	Artemis Urban	**BEV Urban**	BEV Com	FTP	BEV Road	EPA	Artemis Road	WLTC	Artemis Motorway 130	**Artemis Motorway 150**
v max [km/h]	58	61	78	91	93	96	112	131	132	150
Ø v [km/h]	18	21	27	34	57	77	57	46	97	99
time [min]	17	11	13	31	29	26	18	30	18	18
distance [km]	4.9	4.0	5.8	17.8	15.4	33.0	17.3	23.3	28.7	29.5

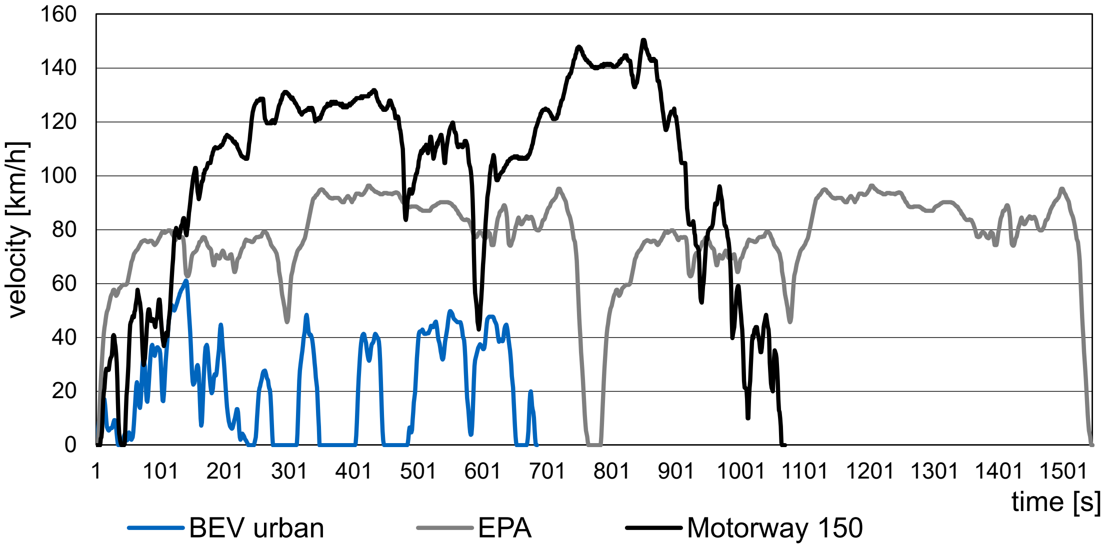
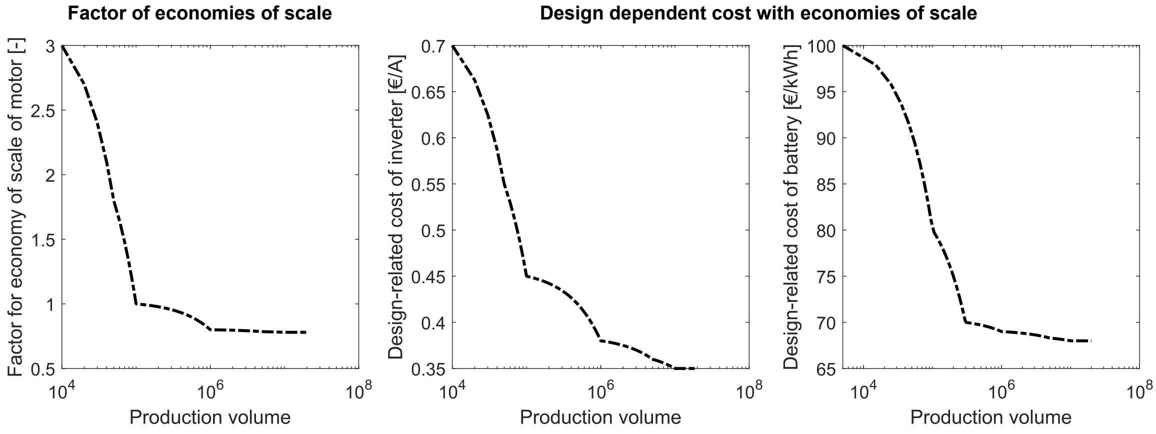


Figure 7.7: Top: overview of driving cycles in the three categories urban, road and motorway. Bottom: examples of three velocity over time curves of three driving cycles.

to their use case scenario: urban, road, and motorway. Three of them are depicted as examples in the diagram at the bottom of figure 7.7. Velocity over time curves describe the cycles. The motorway cycle has the highest velocities and also stays long in areas of high velocity. It has fewer acceleration and deceleration phases than the other two cycles. The road cycle (EPA) has also fewer acceleration and deceleration phases than the urban cycle, but it stays longer in areas of lower velocity than the motorway cycle. Many acceleration and deceleration phases describe the urban cycle. The resulting operating points of the motor (speed and torque) are strongly dependent on the selected driving cycle.

7.1.5 Step 4: Model Cost

The three components which need to be optimized require a cost model. Therefore, the cost functions and parameters from section 6.6 need to be determined.



	Production cost *[€/kg]*		
	low	middle	high
Copper	5.5	6.5	8
Electrical sheet	2	2.5	4
Magnet	35	40	50
Steel	1	1.5	2
Aluminum	1.5	2	2.5

		Production volume		
		low	middle	high
Product	**S**	150 000	250 000	350 000
	M	150 000	250 000	350 000
	L	150 000	250 000	350 000
	XL	50 000	150 000	250 000

		Complexity cost *[€]*		
		low	middle	high
Component	**Motor**	1 000 000	2 400 000	75 000 000
	Inverter	600 000	1 800 000	40 000 000
	Batterie	500 000	1 200 000	30 000 000

Figure 7.8: Top: design-dependent cost function of the three components motor, inverter, battery including effects of economies of scale. Bottom-left: production costs for motor-related parts. Bottom-middle: production volumes of the four product variants for three different scenarios. Bottom-right: complexity costs of the three components for three different scenarios.

Figure 7.8 shows the different elements of the cost model. The diagrams on the upper left part of the figure represent factor of economies of scale for the motor $\hat{d}_{motor}$. The diagrams on the upper middle and upper right incorporate both the function for design-related costs of the component $\hat{c}_{x_j}$ and the function for economies of scale of the component $\hat{d}_j$. For the components inverter and battery the costs can directly be related to the design variables $I_{inverter}$ and E_{bat} respectively. For the motor the the cost scales with the mass of different parts of the motor, which needs to be calculated from the design variables $d = d_{motor}$ and $l = l_{motor}$.

$$\hat{c}_{motor}(d_{motor}, l_{motor}) = 1.4[m_{co}(d,l)c_{co} + m_{stator}(d,l)c_{stator} + m_{mag}(d,l)c_{mag} + m_{rotor}(d,l)c_{steel} + m_{hous}(d,l)c_{alu}] \tag{7.1}$$

Equation (7.1) shows how the function of the design-related costs of the motor $\hat{c}_{motor}(d_{motor}, l_{motor})$. The motors consists of five parts: electromagnetic coils made of copper, stators made of electrical sheet, magnets, rotors made of steel, and housings made of aluminum. The scaling variable of the costs of these parts is the mass. The masses are calculated for each part and then multiplied by the respective production cost value of the part. The costs of each part are then summed up. The masses depend on the design variables of the motor. Figure 7.8 (bottom-left) shows the production costs for each part. 1.4 is a value adding factor.

Together with the production volumes (see figure 7.8 bottom-middle) the design-related costs of the product family $C_{\mathbf{x}}$ can be evaluated.

The complexity costs for each component include expected costs for development, tooling, and administration of component variants. They are given as costs per additional component variant.

Production costs, production volumes, and complexity costs are given in scenarios (*low-middle-high*).

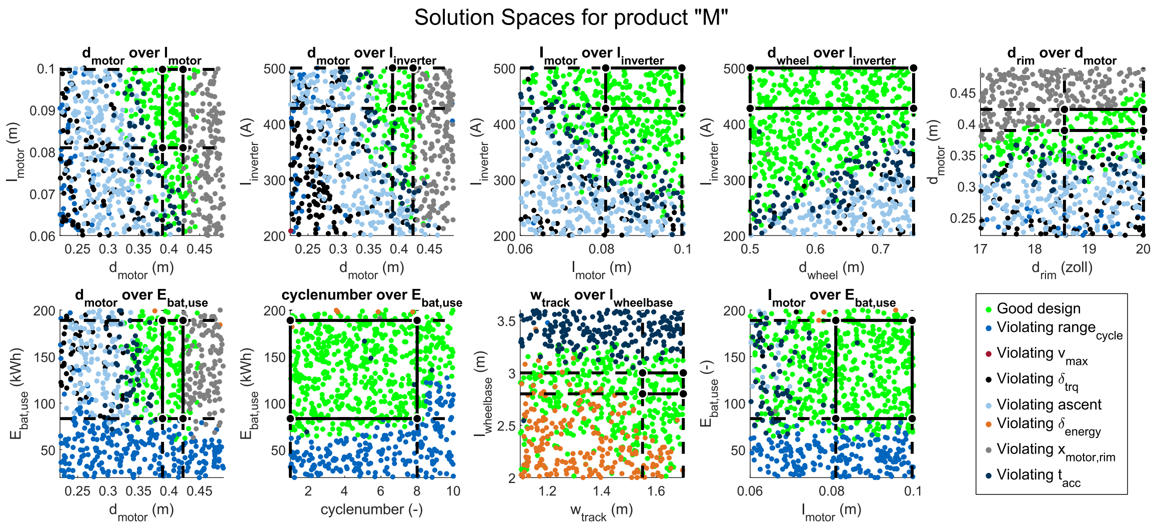


Figure 7.9: Solution spaces of product M. Green dots represent designs which fulfill all requirements. Coloured dots indicate requirement violation.

This is due to the uncertainty the startup is facing. Thus, the data available is subject to uncertainty. This applies for cost data of many companies, but is even more pronounced for a startup.

The cost models and the data for this step are based on knowledge of the experts at the startup and on literature (Ehrlenspiel et al., 2020).

7.1.6 Step 5: Optimize Product Family

Step 5 uses the outputs of the upstream steps.

Product family design using solution spaces (PFD-SSE)
The algorithm samples the design space of the design variables from section 7.1.3. It uses the system model from section 7.1.4 to evaluate the system response, i.e., values for the quantities of interest. The values of the system response for each design point are compared with the requirements from section 7.1.2. If a point fulfills all requirements, it is green. Otherwise it is depicted in the colour of the requirement it is violating. The whole design space consists of 19 design attributes. Some of them are considered as design parameters, such as the friction coefficient μ or energy consumed by auxiliary components p_{aux}.

Figure 7.9 shows maximized solutions spaces for product M. Maximized solution spaces have a maximized rectangular box in the design space. Thus, the intervals for the design variable are maximized. This allows for higher flexibility during design.

The diagram in row one, column one (1-1) shows the solution space for the design variables of the motor d_{motor} and l_{motor}. The interval $[0.389; 0.423]m$ constrains the diameter of the motor d_{motor}. Smaller values of d_{motor} would violate the requirements in the following order: requirement on the acceleration time t_{acc} (dark blue points), *ascent* (light blue points), $range_{cycle}$ (blue points), and ability to provide enough torque to complete cycle at every point δ_{trq} (black points). Higher values of d_{motor} would violate the geometrical requirement on $x_{motor,rim}$ (grey points), i.e., the motor would not fit into the rim anymore. The interval $[0.081; 0.010]m$ constrains the length of the motor l_{motor}. Smaller values would also violate the requirements named above. The upper bound is the limit of the design space.

The diagram in row two, column two (2-2) shows the energy content of the battery $E_{bat,use}$ over the

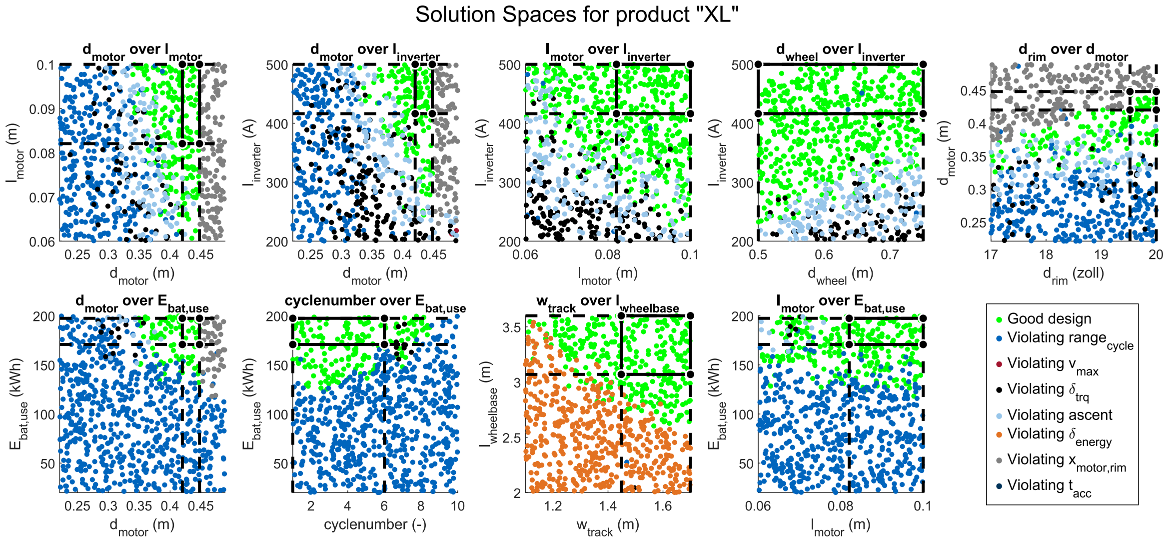


Figure 7.10: Solution spaces of product XL. Green dots represent designs which fulfill all requirements. Coloured dots indicate requirement violation.

cyclenumber, which determines the driving cycle. The design of product M considers cycles one to eight, as indicated by the box-shaped solution space. These are all cycles except the two cycles with high maximum and average velocities (Artemis Motorway 130 & 150, see figure 7.7). For cycles nine and ten, the designs in the box-shaped solution space would not fulfill the range requirement (blue points). The interval $[83.39; 188.84] kWh$ constrains the used energy content of the battery $E_{bat,use}$. Smaller values would violate the range requirement. Higher values would violate the requirement on the maximum energy which can be provided δ_{energy}. This means not enough cells with enough energy content per cell can be integrated in the frame.

This can be also seen in diagram (2-3). It shows the design variables width of track (w_{track}) over the length of the wheelbase ($l_{wheelbase}$). The intervals $[1.55; 1.65] m$ and $[2.8; 3] m$ constrain the design variables respectively. Smaller values of both design variables would violate the requirement on the maximum energy which can be provided δ_{energy}. This means the frame would become too small to include enough battery cells. Higher values for $l_{wheelbase}$ would violate the requirement on the acceleration time t_{acc} (dark blue points). Higher values of $l_{wheelbase}$ reduce the dynamic wheel load on the rear wheel. This reduces the max acceleration force which can be applied on the street, which then limits the acceleration time. The upper limit of the design space constrains the upper limit of w_{track}.

This analysis is done for all products of the product family (S-XL). Figure 7.10 shows the solution spaces for product XL. In contrast to the solution spaces of product M (see figure 7.9), the most crucial QoI is no longer the acceleration time t_{acc} (dark blue dots) but the climbing ability *ascent* (light blue dots) (see diagram (1-1)). The design considers cycles one to six. These are urban and road cycles, as this product represents the class of transporters used within a city and between cities. This diagram also visualizes that the considered set of designs is not suited for driving cycle seven (Artemis Road, see figure 7.7). It cannot provide enough torque to complete the cycle (black dots). Furthermore, the requirement on t_{acc} no longer constrains $l_{wheelbase}$. This is due to the higher loads of the vehicle and the relaxed requirement on t_{acc} (see figure 7.1.2).

The intervals derived from the box-shaped solution spaces constrain the design variables. These intervals contain only good designs. The optimization algorithm uses those intervals to optimize the assignment scheme and the design variable values of the component variants.

Figure 7.11 shows the result of the optimization of the product family using PFD-SSE. The diagram

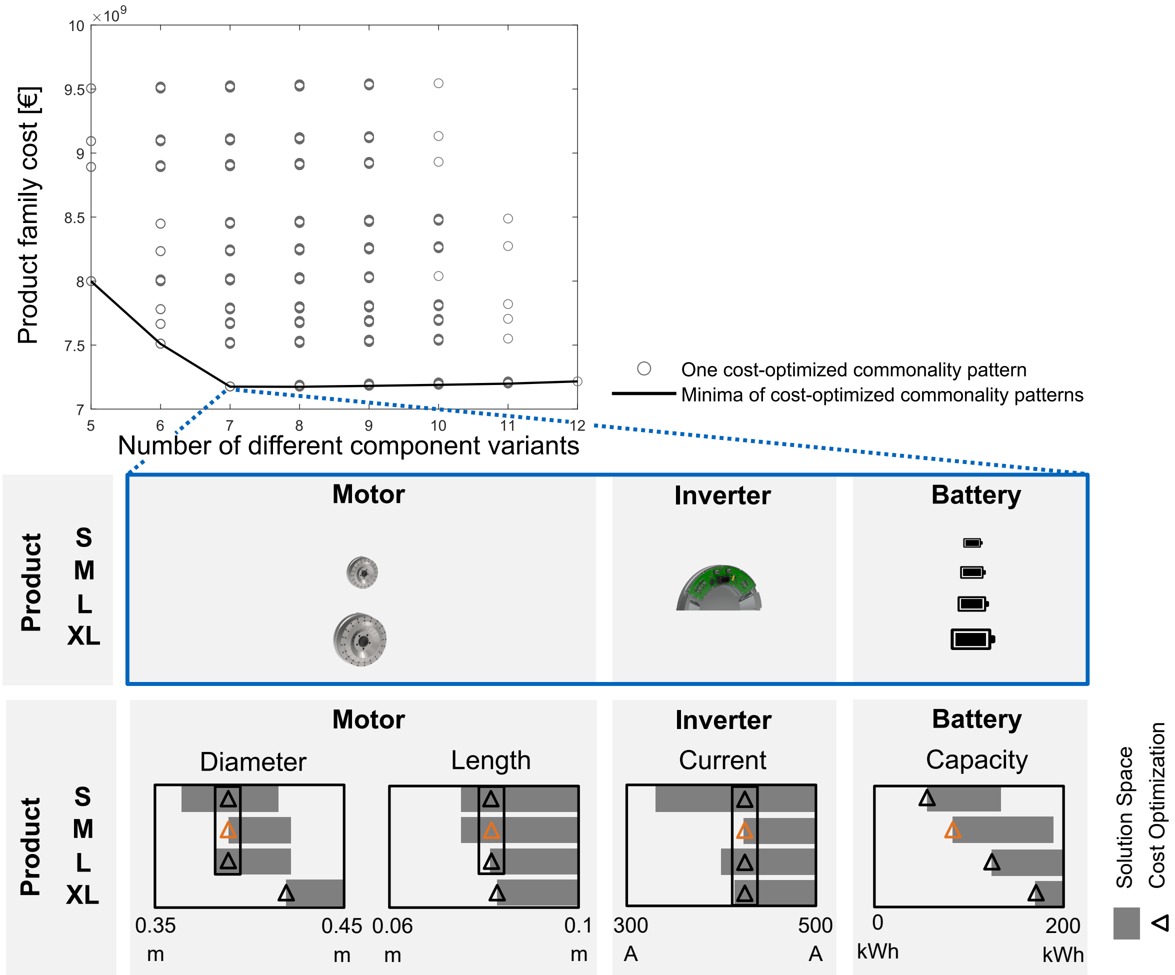


Figure 7.11: *Optimized assignment scheme and design variable values for a product family of motors, inverters and batteries for electric vehicles using PFD-SSE. Top: total cost of the product family over number of components variants. Middle: optimized assignment scheme with components (courtesy of DeepDrive GmbH). Bottom: Optimized design variable values of the component variants. Intervals derived from the box-shaped solution spaces. Triangles represent optimized design variable values. Orange triangles indicate optimized DV values for product M. Figure according to Rötzer et al. (2022a).*

on the top shows the total cost of the product family C_{PF} over the component variants used. It does not distinguish between the type of component variant. It shows all assignment schemes, which are possible from a technical perspective. One dot shows the cost optimized solution for this specific assignment scheme. The black line connects the minima with each other.

In this case the assignment scheme with seven component variants is cost optimal. Figure 7.11 (middle) illustrates the cost-optimized assignment scheme. The seven component variants consist of two motor variants, one inverter variant, and four battery variants. Product variants S, M and L share one motor variant. XL has its own motor variant. All product variants (S-XL) share one inverter variant. Each product variant has its own battery variant, cost-optimized for its use case scenario and requirements.

Figure 7.11 (bottom) shows the optimized assignment scheme and the optimized design variable values for each product variant. The black rectangle indicates that the DV values are same and the corresponding products share this component variant. The grey bars indicate the size of the solution space for the corresponding DV and product variant. They are derived directly from the solution space analysis, as can be seen in figure 7.9 or figure 7.10. They demonstrate the permissible intervals of DVs for each product variant.

In this example higher degrees of commonality are possible from a technical perspective. The four battery variants could be reduced to two battery variants. There are four possibilities for commonality (indicated by '-'): S-M, L-XL; S-L, M-XL; S-M-L, XL; S, M-L-XL. All these possibilities lead to two battery variants. Together with two motor variants and one inverter variant, these two battery variants lead to the highest commonality possible: five component variants. Figure 7.11 (top) shows how the total product family cost rises when the number of battery variants is reduced. A reduction of the number of battery variants comes along with an over-dimensioning of the batteries. As the design-related costs of the battery are high, compared to the other two components, the increase of cost due to over-dimensioning is higher than the savings due to a reduction of component variants. This leads to increasing costs. For this scenario the cost-optimized product family consists of two motor, one inverter and four battery variants. The optimization took less than one minute on a Intel® Core™i7-8550U CPU @ 1.80GHz.

7.1.7 Further Analysis

As explained in section 7.1.1, this example is subject to uncertainty. Uncertainty arises from an economical and a technical perspective. Both can have influence on the cost-optimized design. A scenario analysis of different cost scenarios encounters the economical uncertainty. An analysis of the solution spaces and determination of a maximized commonality space enables robustness from a technical perspective.

Analysis of the economical robustness
The cost model is subject to uncertainty. The startup does not have any information from predecessors. The goal is to identify the influence of different costs scenarios on the optimized product family design. Therefore, the experts at the startup created scenarios for the parameters of the cost model (see section 6.6). They were especially interested in the influence of varying production volumes and complexity costs. There are three scenarios for both parameters *low*, *middle* and *high* (see section 7.1.5).

The factor between the lowest and the highest values of the complexity costs reaches from 60 (battery) to 75 (motor), i.e., introducing an additional motor variant does no longer cost one million euros but 75 million euros. The factor between highest and lowest values of production volumes reaches from 2.3 (S, M, L) to 5 (XL). This means instead of producing 50 000 XL products, the startup produces 250 000. Complexity costs are difficult to determine. Therefore, the uncertainty and the factor is higher.

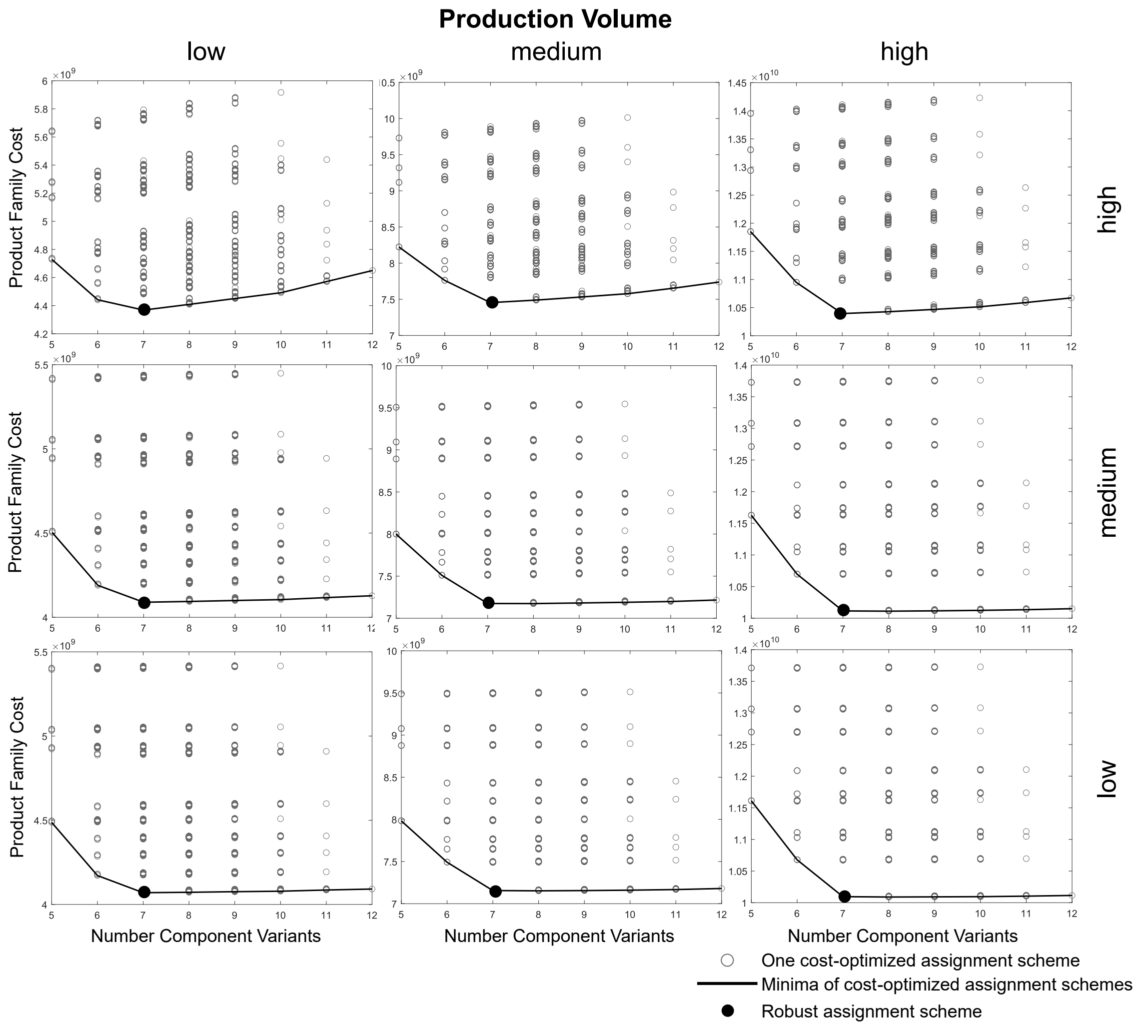


Figure 7.12: Optimization results for different scenarios of complexity costs and production volumes. Robust solution is shown as a black dot. (Rötzer et al., 2022a).

There are nine possibilities to combine the scenarios for the two parameters complexity costs and production volumes. Figure 7.12 shows the nine combinations as a matrix. The rows represent the scenarios of the complexity costs. The columns represent the scenarios of the production volumes. The complexity costs are rising from the bottom to the top of the figure; the production volumes from the left to the right. Each entry represents the optimization results of a specific combination of these scenarios. For example, the diagram in row one - column one (1-1) shows the optimization result of the product family with low production volumes and high complexity costs. The diagram in the middle (2-2) shows the results as shown in figure 7.11.

Each diagram is the optimization result of the product family using different parameters of the cost model (see section 7.1.5). The solution spaces used for the optimization are the same for all combinations. In each diagram a black curve connects the local minima of the different optimized assignment schemes. The overall minimum is marked with a black dot. The curve has to parts: the part to the left of the overall optimum (left part); and the part to the right (right part). The left part includes solutions with lower number of component variants than the optimized solution; the right part the solution with higher numbers.

Two trends can be seen in the diagram. First, increasing complexity costs lead to an increasing slope at the right part of the curve. This means that the overall costs of the product family rise with every additional component variant, which is introduced. This effect pushes the overall optimum to the left, i.e., to less component variants and higher commonality.

Second, increasing production volumes reduce the effect described above. With higher production volumes, the left part of the curve gets steeper and right one flatter. This pushes the overall optimum to the right, i.e., less commonality and more individual components. Commonality comes along with over dimensioning. For high production volumes over dimensioning is disproportionally expensive. It can be economically reasonable to introduce a new component variant. This component variant has to fulfill the requirements of less product variants. Thus, it can be designed more precisely for the considered product variants. Less or no over dimensioning is required and design-related costs can be reduced. This is especially crucial for high production volumes.

The goal of the analysis is to determine the influence of different scenarios on the results, i.e., the optimized assignment scheme and design variable values. With respect to the resulting assignment scheme, there are two combinations, which are crucial: low production volumes and high complexity costs (1-1); high production volumes and low complexity costs (3-3). Diagram 1-1 leads to the highest level of commonality; 3-3 to the lowest. However, in this case there is a solution which is optimal for all combinations (black dot). This solution is robust from an economical perspective. Even if the complexity costs vary by the factor of 75 and the production volumes by the factor of 5, the proposed assignment scheme stays optimal. Each optimization took less than one minute on a Intel® Core™i7-8550U CPU @ 1.80GHz.

Analysis of the technical robustness
The proposed approach from this thesis was applied at a very early phase of product development. So apart from economical uncertainty, there was also technical uncertainty. On the one hand requirements could change, but even more important, the top of the vehicle is not known. The startup provides the rolling chassis, but they do not know yet how their final design will look like (DVs type A) and how potential customers will design the rest of the vehicle (DVs type B).

Design variables of type A are for example: the track width w_{track} or the length of the wheelbase $l_{wheelbase}$. They are not determined yet, but will be during the design process. Design variables of type B are for example: the drag coefficient c_w, the surface of the front of the vehicle A_{front}, the auxiliary power consumption p_{aux}, and the gross weight of the vehicle m_{gross}. They will be influenced by the customers' vehicle design and are still subject to uncertainty when the chassis design is fixed.

Figure 7.13 shows solutions spaces for product M respecting the optimized assignment scheme. The orange triangles indicate the optimized design variable values. Both the optimized assignment scheme and design variable values are from figure 7.11. This includes the following design variables (components): d_{motor}, l_{motor} (motor); $I_{inverter}$ (inverter); $E_{bat,used}$ (battery). In diagrams where only one of these DVs is depicted, the triangle was placed in the middle of the interval of the other design variable. They are not subject to an optimized solution, e.g., in diagram 2-1 d_{wheel} is not part of the optimized components. Thus the triangle is in the middle of its interval.

The diagrams 1-1, 1-2, and 1-3 contain only DVs of optimized components. The box-shaped solution space is the common solution space according to the assignment scheme. This means the box-shaped solution space for diagram 1-1 is the same for the products S, M and L as this three products share the motor. The interval for $I_{inverter}$ is the same for all products of the product family as there is only one component for all of them. The interval for $E_{bat,used}$ is individual for each product. Thus, the triangle for the battery is closest to the limit of bad designs. The other components need to be overdimensioned to be shared. Thus, the rectangle is further away from the limits of bad designs.

The box-shaped solution space indicates the freedom to change the design of the components independently, such that the optimized assignment scheme is still ensured and the requirements of all products are still fulfilled. The designer knows the position of the optimized solution, but can

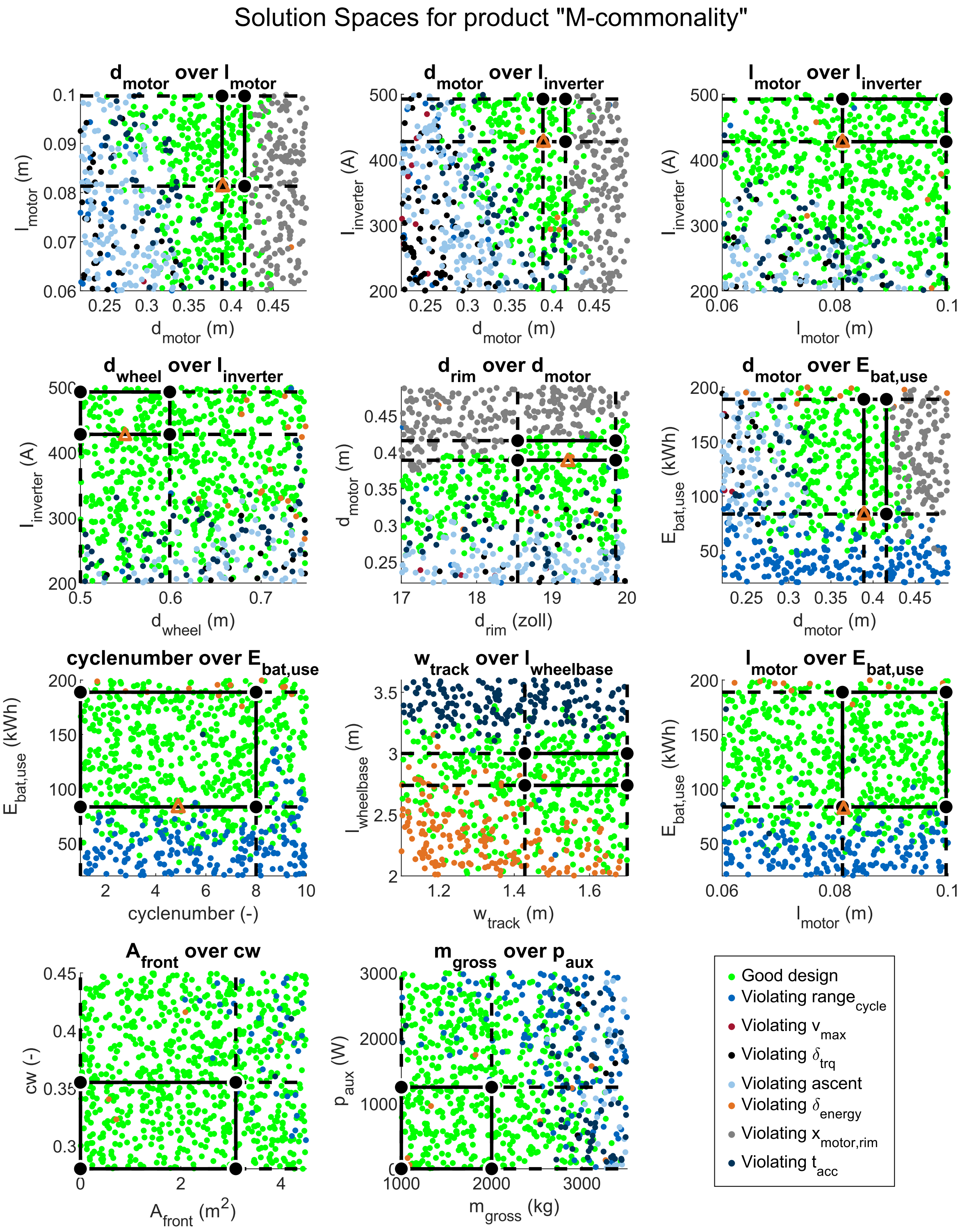


Figure 7.13: Maximized solution spaces for product M respecting the optimized assignment scheme of the product family. Orange triangle indicates optimized design variable values.

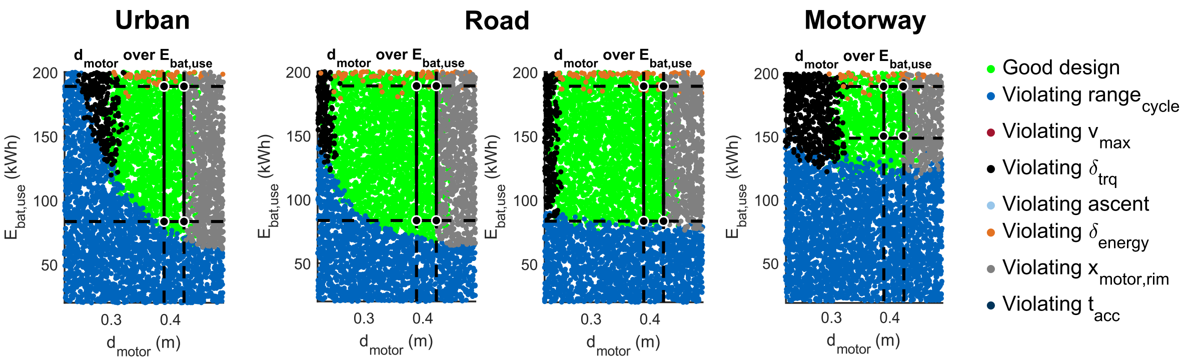


Figure 7.14: Solution Spaces for product M. d_{motor} over $E_{bat,used}$ for different cycles (Rötzer et al., 2022a).

change the values of the DVs to regions in the solution space, which are further away from the limits of bad designs. This increases the robustness of the solution against changing requirements.

Furthermore, the independent intervals for the DVs allow for flexibility during the early phase of the product development process. DVs can be changed within their intervals without changing the overall system. This also allows for robust design.

The design variables which are not part of the optimization process (type A and B) can be constrained such that the optimized assignment scheme and optimized DV values are ensured. For this, the final design does not need to be known. This allows for design for commonality in an early phase.

Diagram 3-2 of figure 7.13 shows DVs of type A. Intervals constrain the track width and the length of the wheelbase. A designer can choose values for the DVs within these intervals independently. They know that, as long as the values stay within the respective intervals, the optimized assignment scheme, the optimized DV values and all requirements can be achieved.

Diagrams 4-1 and 4-2 show DVs of type B. They will be determined by a future customer. However, they influence the behaviour of the overall system. Thus, they need to be constrained as well. The larger the intervals, the less strict the constraints need to be. The considered components can be designed although not all variables are known yet. The intervals allow for flexibility and robustness, such that the overall system does not need to be changed.

Analysis of the interaction between the cycles and design
Another interesting aspect in this evaluation case is the interaction between the cycles and the design. The performance of a vehicle is subject to its field of application. This adds another dimension of complexity. The cycles from figure 7.7 represent different application scenarios. They determine in which operation points the motor works. The design of the motor on the other hand determines in which operation points the motor works efficiently (see figure 7.6). Together with other variables, such as the drag coefficient, the front surface or the gross weight, they determine the QoIs, such as the range.

Figure 7.14 shows the influence of different cycles on the design of the battery ($E_{bat,used}$) and the motor (d_{motor}). There is one urban cycle, two road cycles, and one motorway cycle. Two observation can be made: First, the minimum level of the blue dots ($range_{cycle}$) moves upwards. Thus, the energy content of the battery increases from left to right (urban - road - motorway). This means for instance that more energy is needed in a motorway cycle than in a road cycle.

Second, the limit line of blue dots gets flatter from left to right. This results from the influence of the motor design (here represented by d_{motor}) on the range. Many acceleration and deceleration sequence dominate the urban cycle. High and rather constant velocities characterizes the motorway cycle (see figure 7.7). The road cycles can be something in between. Here two cycles are shown as an example. The BEV Road cycle with more acceleration parts (road-left) and EPA with higher and more constant velocities (road-right). For the cycles with fewer acceleration sequences the motor design has a lower impact. The almost horizontal line of blue dots in diagram three and four show that d_{motor} has little influence. In these cases, the rolling and, especially in the motorway cycle, the air resistance dominate the power consumption and thus the range. For smaller velocities with more acceleration sequences (diagram one and two) the losses of the drivetrain are more dominant. Especially in the urban use case, the influence of d_{motor} is highly non linear. This dependency results from the influence of the motor diameter on the engine maps. With smaller motor diameters, the operating points are at (relatively) higher loads. There the motor is less efficient (see figure 7.6). By considering the driving cycles typical use case scenarios for different product variants can be taken into consideration for the design of the product family.

7.2 Evaluation Case 2: Water Hose Box Design

This evaluation case describes the results of a cooperation with a consulting firm. Parts of the results presented here have been published in Rötzer et al. (2020b), Ehrlenspiel et al. (2020), Rötzer et al. (2021) and Rötzer et al. (2022b).

7.2.1 Introduction

A manufacturer of gardening tools commissioned a consulting firm to re-design their product portfolio of water hose boxes. The manufacturer was interested in an optimal degree of standardization of the components within the product family.

The water hose boxes store a water hose which can be used to water a garden. The user can pull the water hose out of the box. After the usage, the box can pull back the water hose automatically by a spring. The water hose boxes consist of three components: reels, springs, and housings. The mechanism which locks the reel when the hose is pulled out and the mounting of the box is not considered here.

Figure 7.15 (left) shows a sketch of a garden hose box on a mounting (not considered here). The sectional view in the middle (A-A) illustrates the components of the water hose box and their design

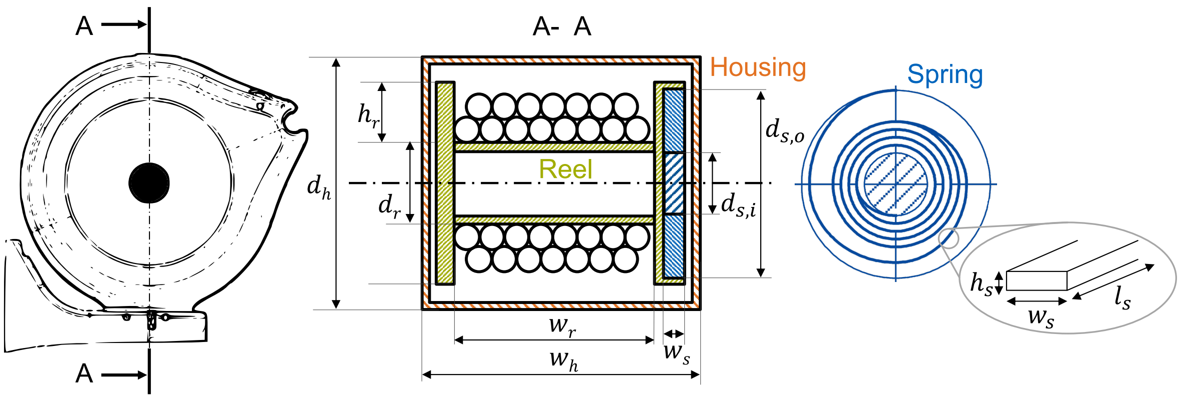


Figure 7.15: Left: Sketch of the water hose box. Middle: Sectional view of the water hose box with components. Right: Detailed view of the spring.

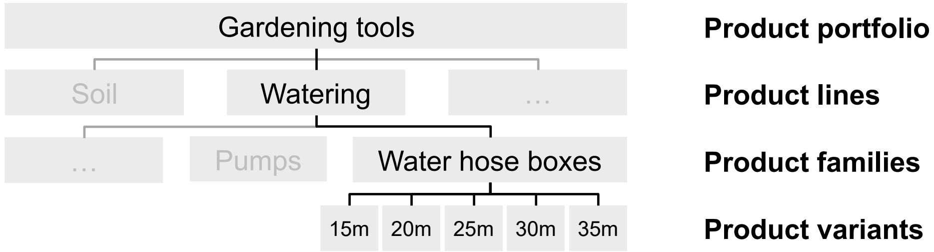


Figure 7.16: Product portfolio of the manufacturer of gardening tools with product family of garden hose boxes.

variables. The hose is rolled up on the reel in horizontal and vertical layers. When the hose is pulled out, it winds the spring. The energy is stored in the spring to pull back the hose again. The spring consists of spring steel sheet, which is mounted and coiled up. All components are stored within the housing.

Figure 7.16 shows the product portfolio and the considered product family of the gardening tools manufacturer. The water hose box should be offered on the market in five variants. The product variants differ from each other according to the hose length they can store: 15m, 20m, 25m, 30m and 35m.

Table 7.2 shows that the example meets the prerequisites of the developed approach.

ID	Prerequisite
PR 1	It is a scale-based product family. The components reel, spring, and housing are part of all product variants, but can be scaled differently.
PR 2	Requirements on the product variants are quantitative, such as the hose length, the ratio of housing, or the insertion forces.
PR 3	Product variants differ from each other by different requirements on hose length and max. insertion forces.
PR 4	Design variables are quantifiable, i.e., the component variants differ from each other by different dimensions of the reel, spring, and housing.
PR 5	System behaviour is quantifiable. Formulae can describe the winding of the hose. Spring design tools can calculate the spring characteristics from the dimensions of the spring.

Table 7.2: Prerequisites of the developed approach for evaluation case 2

7.2.2 Step 1: Collect Requirements

The customer-relevant requirements come from the gardening tools manufacturer. They are augmented with requirements which are needed for the functioning of the products.

Figure 7.17 shows the requirements of the product family of water hose boxes. The first requirement on the hose length is the one which is determining the variants. It differs from product variant to product variant.

Product variant	Hose length [m]	Ratio of housing [-]	Insertion force min [N]	Insertion force max [N]	*Δd* spring /reel [m]	*Δd* reel/ housing [m]	*Δw* reel/ housing [m]	No. min. spring turns [-]	Bending of hose [-]
15m	≥15	0.5 – 0.6	≥10	≥21	≥0.01	≥0.03	≥0.04	≥7	≥14
20m	≥20	0.5 – 0.6	≥10	≥28	≥0.01	≥0.03	≥0.04	≥7	≥14
25m	≥25	0.5 – 0.6	≥10	≥35	≥0.01	≥0.03	≥0.04	≥7	≥14
30m	≥30	0.5 – 0.6	≥10	≥43	≥0.01	≥0.03	≥0.04	≥7	≥14
35m	≥35	0.5 – 0.6	≥10	≥50	≥0.01	≥0.03	≥0.04	≥7	≥14

Figure 7.17: Quantities of interest and requirements for the product variants of the product family of water hose boxes

The ratio of the housing constrains the ratio between the width and the diameter of the housing for aesthetic reasons. The min and max value of the insertion force need to reach at least certain values to pull back the hose. The min value applies when the hose is pulled in; the max value when it is pulled out completely. The three requirements indicated with a Δ describe gaps. They are needed so that the components fit into each other: the spring into the reel, the reel and the spring into the housing. The spring needs a certain preload to work in its operating range. This is achieved by a minimum number of spring turns. This requirement ensures that after the spring is released, there is still a certain number of turns coiled up. This implicitly also ensures that there are enough turns to pull back the hose completely. If the hose is bent too much, it can buckle. To prevent this, a minimum ratio between the inner diameter of the reel and the diameter of the water hose is required.

7.2.3 Step 2: Model System Dependencies

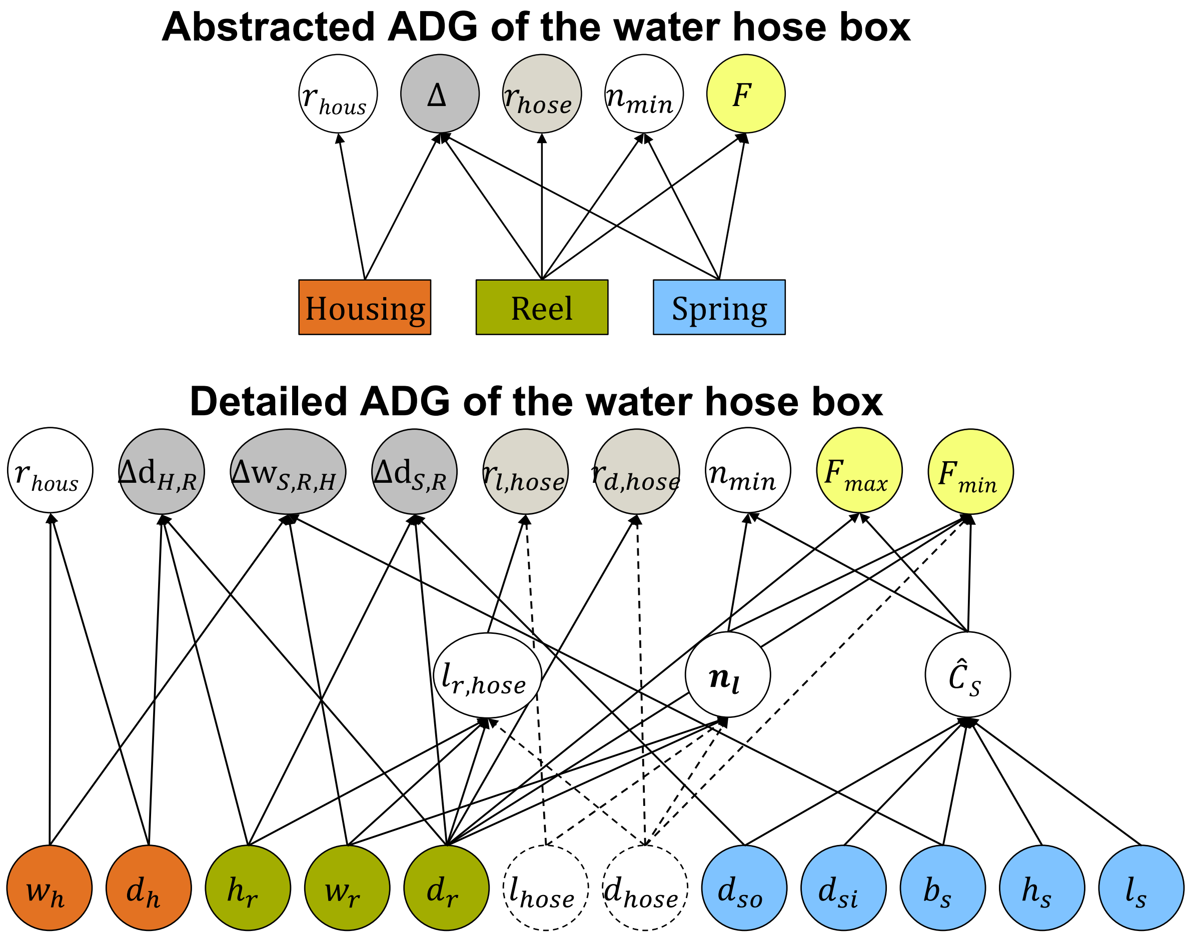
Figure 7.18 shows the dependency structure of the water hose box as an attribute dependency graph (ADG). The abstracted ADG at the top of the figure provides an overview of the components and how they influence the QoIs. Both the DVs and the QoIs are grouped. The DVs are grouped according to the component they belong to. The abstracted ADG has the following QoIs: the ratio of width and diameter of the housing r_{hous}; all gaps between components, summarized as Δ; QoIs regarding the water hose r_{hose}; number of spring turns n_{min}; and the insertion force F. The DVs of the housing influence the housing ratio r_{hous} and the gaps Δ. The DVs of the reel influence the gaps Δ, the QoIs related to the hose r_{hose}, the spring turns n_{min} and the insertion force F. The spring influences the gaps Δ, the spring turns n_{min} and the insertion force F.

The detailed ADG at the bottom of figure 7.18 allows for a in-depth analysis of the dependencies. The DVs and QoIs have the same colour as the aggregated attributes at the top of the figure. The sketch of the water hose box (figure 7.15) also shows the DVs with the corresponding colours.

The DVs of the spring constitute a certain spring characteristic $\hat{C}_s$. This spring characteristic then influences the resulting forces (F_{max}, F_{min}) and the minimal turns (n_{min}).

The length of the hose l_{hose} and its diameter d_{hose} are parameters in this design task. They influence other attributes but are already fixed by the manufacturer. Therefore, those attributes and their dependencies are marked with dashed lines. The corresponding QoI is $r_{l,hose}$. $r_{l,hose}$ ensures that the hose length, which results from the reel design ($l_{r,hose}$) is at least as big as the hose length assigned to the product variant l_{hose}.

The diameter d_{hose} and the length of the hose l_{hose} together with the width of the reel w_r and the diameter of the reel d_r determine the number of layers n_l which are stored in the reel. The overall number of layers equals the turns, which the spring needs to perform to pull back the hose. Thus, it



DV	Description
w_h	Housing width
d_h	Housing diameter
h_r	Reel height
w_r	Reel width
d_r	Reel diameter
l_{hose}	Prescribed hose length
d_{hose}	Diameter of hose
d_{so}	Outer diameter spring
d_{si}	Inner diameter spring
b_s	Width of spring coil
h_s	Height of spring coil
l_s	Length of spring coil

QoI	Description
r_{Hous}	Side ratio housing
$\Delta d_{H,R}$	Gap between housing and reel diameter
$\Delta w_{S,R,H}$	Gap between housing and reel + spring width
$\Delta d_{S,R}$	Gap between reel and spring diameter
$r_{l,hose}$	Difference between prescribed and realized hose length
$r_{d,hose}$	Diameter ratio d_r to dh_{ose}
n_{min}	Pre-load turns of the spring
F_{Max}	Max. retraction force
F_{Min}	Min. retraction force

Figure 7.18: Attribute dependency graphs (ADG) of the water hose box. Top: Abstracted ADG providing an overview of attributes and dependencies. Middle: Detailed ADG with intermediate attributes. Bottom: Description of design variables and quantities of interest. Figure according to (Rötzer et al., 2022b).

influences n_{min}. n_l also includes the number of vertical layers. The number of vertical layers influences the minimal insertion force F_{min}.

The insertion force applied on the hose is split into its maximal value F_{max} and its minimal value F_{min}. F_{max} occurs when the hose is pulled out completely and the spring is tensioned to the maximum. At the same time the leverage is minimal as the whole hose is pulled out. This leads to the maximum force F_{max}. The minimum force F_{min} occurs when the hose is completely pulled up. Then the tension of the spring is reduced and the leverage is maximum. Thus, the insertion force is minimal.

The attribute Δ describes the gaps between the components. Requirements on these gaps ensure that the components do not overlap. Using gaps allows for an independent design of the components. If there was a dependency for example between the spring reel diameter and the housing diameter, such that the reel determines the housing, it would be impossible to standardize the housing for different reels. This way of modelling allows for additional degrees of freedom without violating the requirements of the product variants.

7.2.4 Step 3: Model System Behaviour

In this step, the dependencies from the ADG in section 7.2.3 are quantified. Each arrow needs to be part of a model. Figure 7.19 shows all the modular models used to describe the system behaviour with their ID. In total the algorithm, presented in section 6.5, merges eight modular models to create the system model.

To provide a better overview, those eight models are grouped into four categories: housing, gaps, spring functions, and hose functions. Each group has a specific colour which highlights the models in the ADG. This assignment illustrates how the different models interact with each other.

The *housing* model ensures a certain ratio between diameter and width of the housing. *Gap* models ensure that the components do not intersect.The *spring functions* calculate the spring characteristics from its design variable values and the resulting forces with respect to the number of layers of the hose. The *hose functions* calculate the number of hose layers in the reel and the length of the stored hose.

The following modular models represent parts of the system model.

The modular model *M0108* calculates both the realized hose length of the reel design $l_{r,hose}$ and the number of layers n_l. n_l includes the number of horizontal layers $n_{l,v}$ and the sum of all layers $n_{l,sum}$.

Figure 7.20 shows the model of winding the water hose on the reel. The water hose is winded such that the following horizontal layers lie between the previous ones.

Equation (7.2) calculates the horizontal distance between two vertical layers.

$$h = \sqrt{3}\,\frac{d_{hose}}{2} \tag{7.2}$$

The algorithm in iterates over all vertical layers and checks how many horizontal layers fit into the reel width. Equation (7.3) calculates the length of the hose which was rolled up on one specific horizontal and vertical layer $i_{l,v}$.

$$l_{r,hose,i}(i_{l,v}) = \pi(d_r + d_{hose}(1 + (i_{l,v} - 1)\sqrt{3})) \tag{7.3}$$

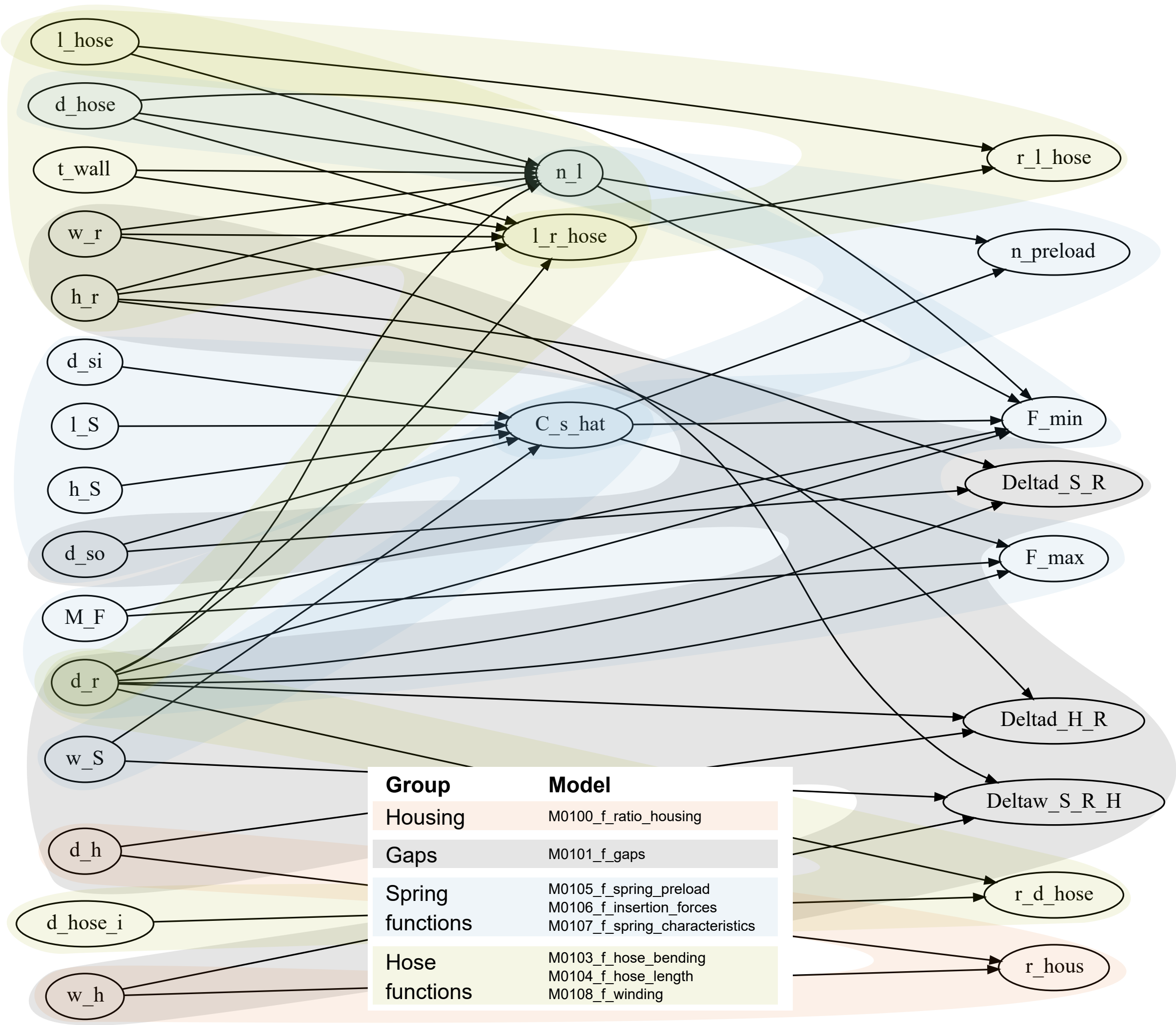


Figure 7.19: Modular models of the water hose box integrated in an attribute dependency graph (ADG).

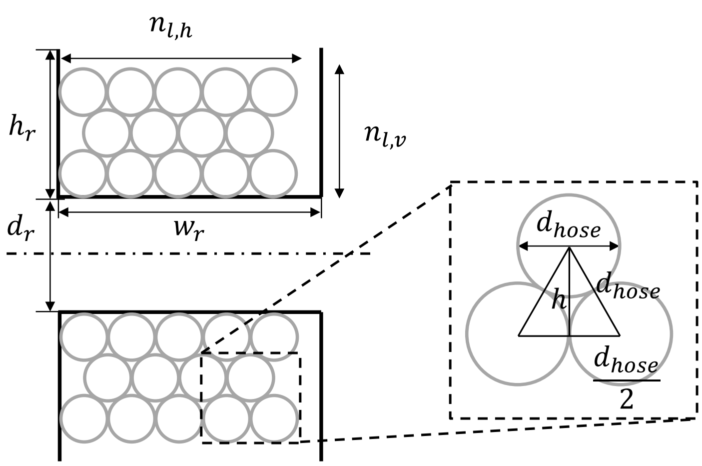


Figure 7.20: Model of winding the water hose

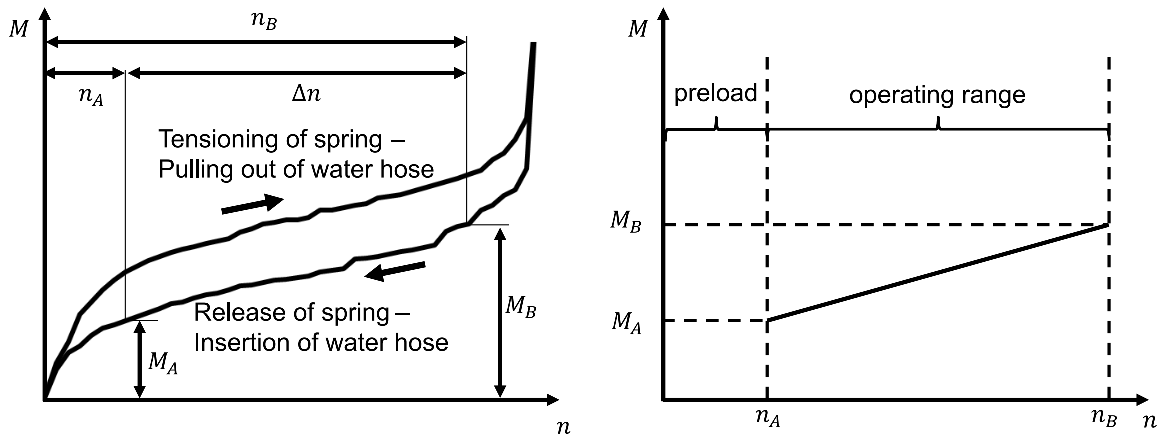


Figure 7.21: Spring characteristic ($\hat{C}_s$). Left: Exemplary characteristic curve of a rotational spring. Right: Linearized operating range during the insertion of the water hose used for the design of the family of water hose boxes

The length which can be rolled up increases with each additional vertical layer by $\pi\sqrt{3}d_{hose}$. It is summed up first over all horizontal layers, and second, over all vertical layers. It saves the number of vertical layers $n_{l,v}$ and the sum of all layers $n_{l,sum}$ when the value of l_{hose} is reached. It calculates the hose length which can be realized with the spring design by continuing until h_r is reached.

The most complicated part of the system model is the calculation of the spring characteristics (*M0107*). Figure 7.21 (left) shows an exemplary characteristic curve (torque M over rotations n) of a rotational spring used in the water hose boxes.

It is a hysteresis curve due to the friction between the layers of the spring. This makes the calculation of the spring characteristics from the design variables of the spring (see figure 7.15) challenging. The coils of the spring steel sheet contact each other depending on the number of rotations. The modular model needs to be able to compute a spring characteristic from the design variables of the spring. For that purpose a software tool for spring design was used. But the calculation time of the spring software was too high to be used directly in the approach.

A neural network mitigates this disadvantage. At first, the characteristic curve is reduced to that part, which is relevant for the problem at hand: the insertion of the water hose box. This is the lower part of the hysteresis curve. Within the lower part there are three parts: two highly non linear ones and one approximately linear one. The non-linear ones are excluded from further consideration. The linear part is the operating range during the insertion of the water hose. It is defined by the points A and B. Figure 7.21 (right) shows the simplified model used for the product family design of the water hose boxes. All values between A and B are interpolated linearly. The B values define the maximum values of rotations and torque provided by a certain spring design. Point A defines the corresponding minimum values. The requirement on n_{min} ensures that the solutions stay within the limits of the model. Equation (7.4) shows how n_{min} is calculated.

$$n_{min} = n_B - n_{l,sum} \tag{7.4}$$

The spring design software calculates the values of n and M at point A and B for different values of the design variables. They are altered within the limits of the design space. This data trains a neural network, which calculates n_A, n_B, M_A, and M_B for arbitrary spring designs within the limits of the design space. A linear interpolation calculates values of $M_{min} = M(n_{min})$, which lie between point A and B.

Component	Production cost
Reel	8 000 €/m^3
Housing	16 €/m^2
Spring	1.5€ + 20 000 €/m^3

Component	Complexity cost [€] low	middle	high
Reel	2 000	10 000	50 000
Housing	15 000	25 000	40 000
Spring	500	1 000	10 000

Product	Production volume
15m	8 000
20m	9 000
25m	10 000
30m	10 000
35m	10 000

Component	Discount rate [-] low	middle	high
Reel	0.02	0.05	0.08
Housing	0.02	0.08	0.10
Spring	0.01	0.02	0.08

Figure 7.22: Parameters of the cost model for the design of a family of water hose boxes. Top-left: design-related production costs of the three components. Top-right: complexity costs of the three components for three different scenarios. Bottom-left: production volumes of the five product variants. Bottom-right: discount rates for economies of scale of the three components for three different scenarios.

M0106 calculates the insertion forces. Equation (7.5) calculates the minimal insertion force. It uses the torque (M_{min}) and the lever at the point when the hose is coiled completely ($i_{l,v} = n_{l,v}$). Equation (7.6) calculates the maximal insertion force. It uses the torque ($M_{max} = M_B$) and the lever at the point when the hose is pulled out completely ($i_{l,v} = 1$). Equation (7.7) calculates the lever according to the number of vertical layers ($i_{l,v}$). The number of vertical and horizontal layers can only be an integer value: either the water hose fits in or not. It is not reasonable to allow fractions of a water hose. This leads to discontinuous system responses.

$$F_{min} = M_{min}/r(i_{l,v} = n_{l,v}) \tag{7.5}$$

$$F_{max} = M_{max}/r(i_{l,v} = 1) \tag{7.6}$$

$$r(i_{l,v}) = \frac{1}{2}\left(d_r + d_{hose}(1 + (i_{l,v} - 1)\sqrt{3})\right) \tag{7.7}$$

Thicknesses of the reel and the housing are also considered, but set as a parameter during this calculation. For a better overview they are not shown here.

7.2.5 Step 4: Model Cost

The three components housing, reel, and spring require a cost model to be optimized. Therefore, the cost functions and parameters from section 6.6 need to be defined.

Figure 7.22 shows the parameters of the cost model used for the optimization of the product family of water hose boxes. It includes production costs (top-left) which scale with volume (reel and spring) and surface (housing); the complexity costs (top-right) of the components for three different scenarios; the production volumes (bottom-left) of the five product variants; and the discount rates (bottom-right), which are used to calculate the effects of economies of scale (see section 6.6).

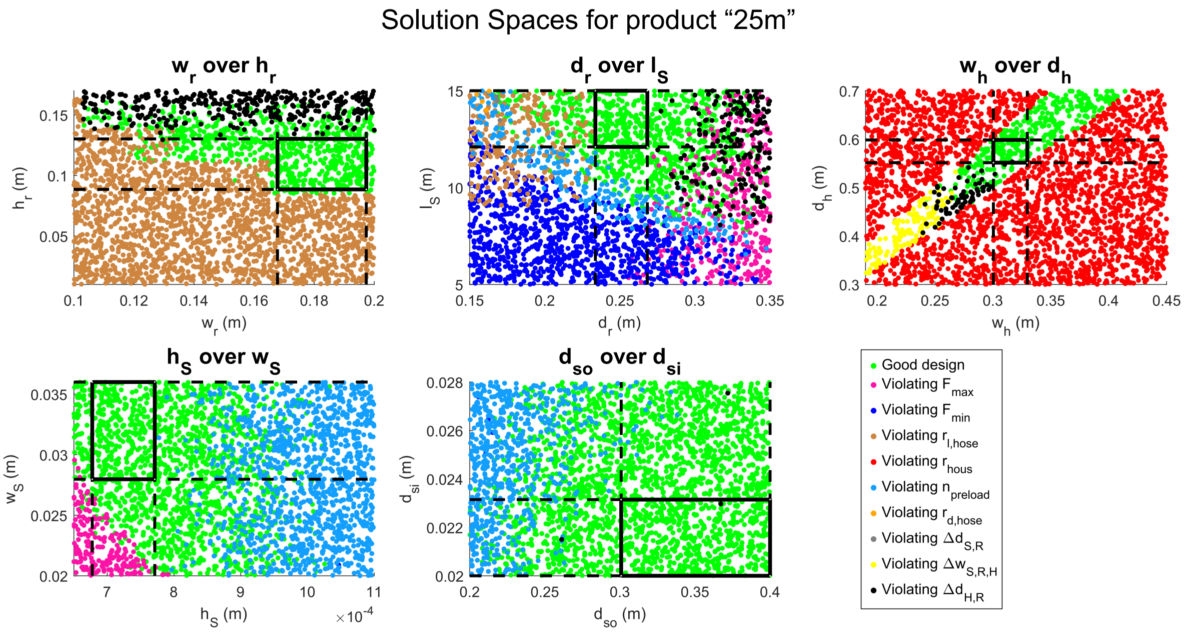


Figure 7.23: Solution spaces of product 25m. Green dots represent designs which fulfill all requirements. Coloured dots indicate requirement violation.

The design-related cost models scale with material volume or surface. This correlation appeared while analyzing the cost data of existing components. Increase of cost according to surface or volume are typical for many products (Ehrlenspiel et al., 2020). The volumes and the surface can be derived directly from the geometrical design variables of the components.

The cost models and the data for this step are based on cost data of existing components, knowledge of the experts at the consulting firm, and on literature (Ehrlenspiel et al., 2020).

7.2.6 Step 5: Optimize Product Family

In this example both algorithms are applied: product family design using solution spaces (PFD-SSE) and two level cost optimization of product families using the complete design space (PFD-CDS). Both algorithms base on the outputs of the previous steps.

Product family design using solution spaces (PFD-SSE)

The algorithm samples the design space of the design variables from section 7.2.3. It uses the system model from section 7.2.4 to evaluate the system response, i.e., values for the quantities of interest. The values of the QoIs for each design point are compared to the requirements from section 7.2.2. If a point fulfills all requirements, it is green. Otherwise it is depicted in the colour of the requirement it is violating. The whole design space consists of ten design variables. Design parameters, such as, water hose length and diameter or wall thickness of reel and housing are not subject to optimization. Nine QoIs describe the system behaviour. Box-shaped solution spaces decouple the design variables from each other and are used as an input for the optimization algorithm. The optimization algorithm uses the cost model from section 7.2.5 as an objective function.

Figure 7.23 shows the solutions spaces for product 25m. The diagram in row one, column one (1-1) shows the solution space for the design variables of the reel w_r and h_r. The interval $[0.168; 0.197]m$ constrains the width of the reel w_r. Smaller values of w_r would violate the requirement on the hose length $r_{l,hose}$ (brown points). The limits of the design space constrain higher values of w_r. The interval $[0.088; 0.130]m$ constrains the height of the reel h_r. Smaller values would also violate the requirement on $r_{l,hose}$. Higher values of h_r would violate the requirement on the gap between reel and housing (black points). The step curve in this diagram also visualizes the discontinuous character of this design problem. The realizable hose length can only increase when an entire layer can be added horizontally or vertically. As long as this is not possible, an increase of w_r or h_r does not result in a larger realizable hose length. But as soon as a new layer can be added, the curve drops by leaps and bounds.

The diagram in row one, column three (1-3) shows the width w_h and the diameter d_h of the housing. Red points indicate that the ratio between diameter and width is violated (r_{hous}). As the requirement on r_{hous} has a lower and upper bound the limits of the solution space have the shape of a corridor. The black points indicate that the gap requirement is violated, i.e., the housing would be too small for the other components.

The DVs of the spring (l_s, h_s, w_s, $d_{s,i}$, $d_{s,o}$) of this product variant are mostly constraint by the requirements on the minimum number of spring turns (dark blue points) and on the max insertion force (pink points).

This analysis is done for all products of the product family (15-35m). The intervals derived from the box-shaped solution spaces of the different product variants constrain the design variables. These intervals contain only good designs. PFD-SSE uses those intervals to optimize the assignment scheme and the design variable values of the component variants.

Figure 7.24 shows the result of the optimization of the product family using PFD-SSE for the ‘middle’ values of the cost model (see figure 7.22). The diagram on the top shows the total cost of the product family C_{PF} over the number of component variants used. It does not distinguish between the types of component variants. It shows all assignment schemes, which are possible from a technical perspective. One dot shows the cost optimized solution for one specific assignment scheme. The black line connects the local minima with each other.

In this case, the assignment scheme with ten component variants is cost optimal. Figure 7.24 (middle) illustrates the cost-optimized assignment scheme. The ten component variants consist of three housing variants, two reel variants, and five spring variants. Product variants 15 and 20m share one housing variant. The product variants 25 and 30 share another housing variant. Product variant 35m has its own housing variant. Product variants 15m and 20m share one reel variant. The product variants 25, 30 and 35 share another reel variant. Each product variant has its own spring variant, cost-optimized for requirements of each product variant.

Figure 7.24 (bottom) shows the optimized assignment scheme and the optimized design variable values for each product variant. The black rectangles indicate that the DV values are same and the corresponding products share this component variant. The grey bars indicate the size of the solution space for the corresponding DV and product variant. They are derived directly from the solution space analysis, as can be seen in figure 7.23 for the 25m variant. They demonstrate the permissible intervals of the DVs for each product variant.

In this example higher degrees of commonality are possible from a technical perspective. The five spring variants could be reduced to two spring variants. The three housing variants could be reduced to two housing variants. Together with two reel variants, they lead to the highest commonality: six component variants. Figure 7.24 (top) shows how the total product family cost rises when the number of component variants is reduced. The reduction of housing variants by over-dimensioning of the 25m-30m variant would lead to a moderate increase in cost (see curve from ten to nine component variants). A reduction of the number of spring variants comes also along with an over-dimensioning of the springs. As the design-related costs of the spring are high, compared to the other two components, the increase of cost due to over-dimensioning is higher than the savings due to a reduction of component variants. This leads to increasing costs.

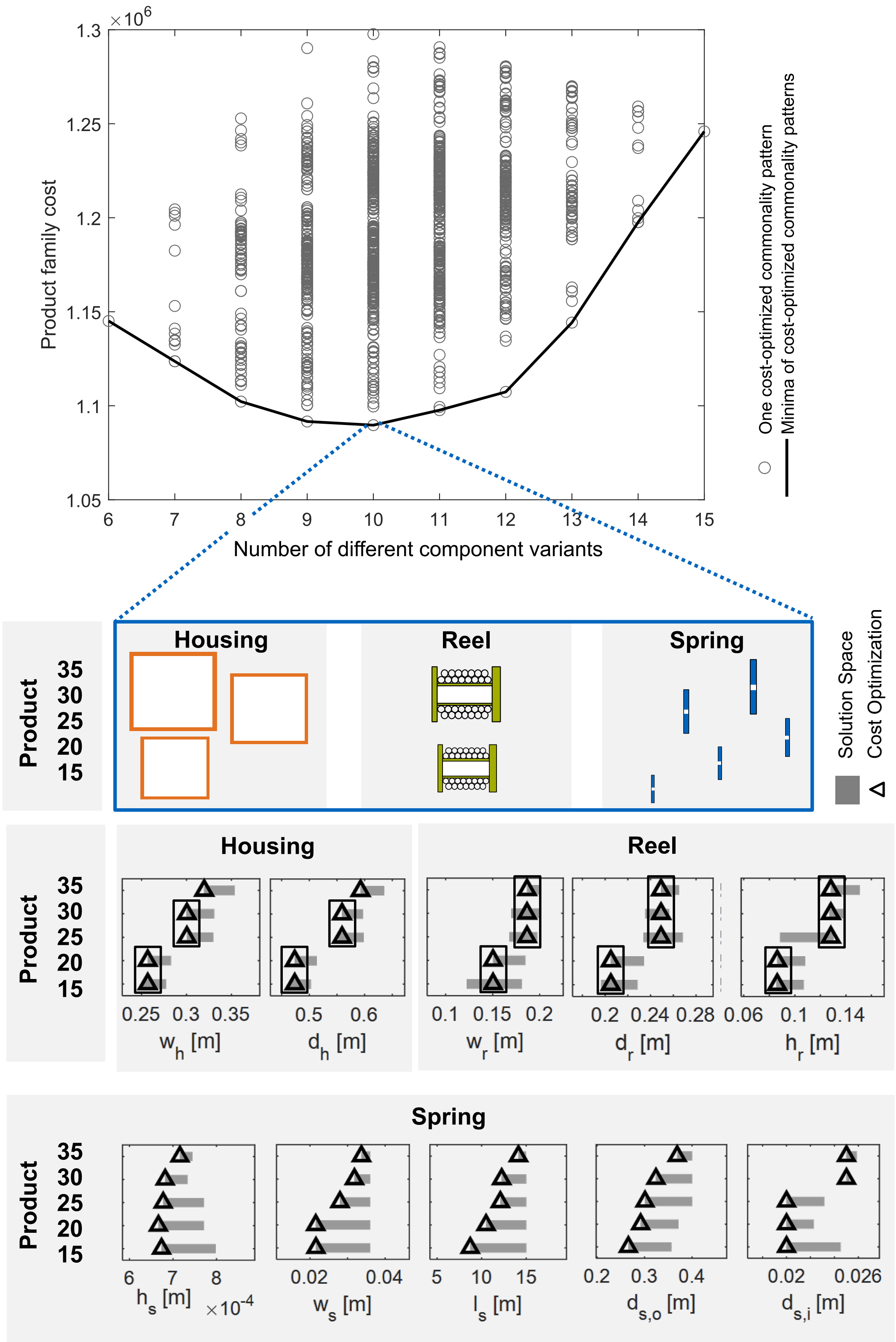


Figure 7.24: Optimized assignment scheme and design variable values for a product family of water hose boxes using PFD-SSE. Top: total cost of the product family over number of components variants. Middle: optimized assignment scheme with components. Bottom: Optimized design variable values of the component variants. Intervals derived from the box-shaped solution spaces. Triangle indicates optimized design variable values.

Larger numbers of component variants lead to higher complexity costs and less effects on economies of scale. The highest number of component variants is 15 (five product variants times three component variants).

The optimization took 12s on an Intel® Core™i7-10700K CPU @ 3.80GHz.

Two level cost optimization of product families using the complete design space (PFD-CDS)

PFD-CDS optimizes the design problem using the complete design space. Figure 7.25 shows the results of PFD-CDS. The diagram on the top shows the total cost of the product family C_{PF} over the component variants used.

Figure 7.25 (middle) shows the cost optimized commonality according to PFD-CDS. The cost optimized product family consists of six component variants: one housing, one reel, and four springs. The product variants 15m, 20m, and 25m have their own spring. The product variants 30m & 35m share a spring variant. The PFD-CDS algorithm is limited to commonality between neighbouring product variants only (see equation (6.8)). For this evaluation case, this is a reasonable assumption: It would be neither economically nor technically reasonable if, e.g., the 15m and 25m hose box had a common reel/housing/spring, while the reel/housing/spring for the 20m water hose box was different. This would mean highly over-dimensioning the components for the 15m water hose box, while the 20m one could be smaller than the 15m one.

Figure 7.25 (bottom) shows the cost optimized design variable values according to the optimized commonality.

The diagram on the top shows that higher levels of commonality are possible (four component variants). But this would lead to higher costs due to over-dimensioning. Higher numbers of component variants also lead to increased costs due to increased complexity costs and less economies of scale. This diagram does not show all possible commonality patterns as a genetic algorithm was used to determine the optimized commonality. Therefore, there can be jumps in the line of local minima, which may occur due to points that the algorithm did not consider.

The optimization took 55 minutes on an Intel® Core™i7-10700K CPU @ 3.80GHz.

7.2.7 Further Analysis

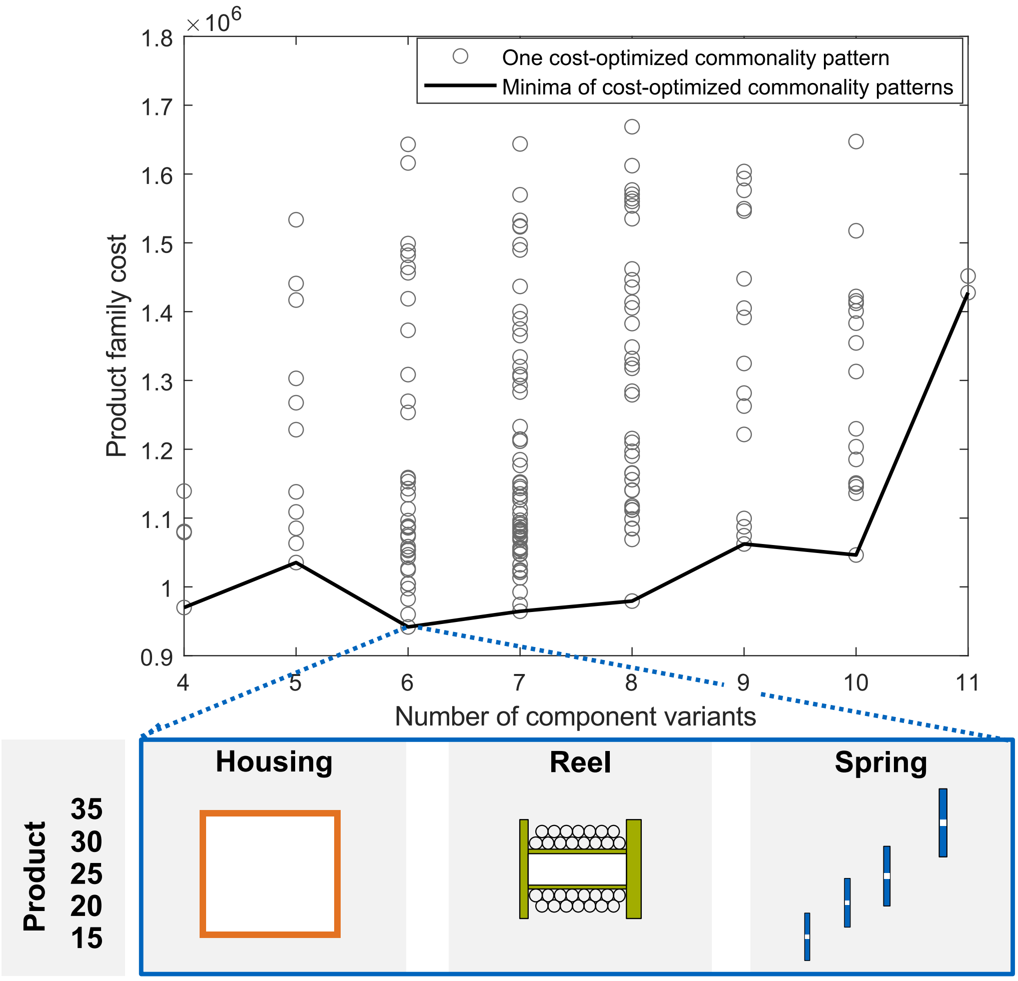
Comparison of PFD-SSE and PFD-CDS solution

The overall product family cost of the PFD-CDS solution is around 14% lower than the one of the PFD-SSE solution. The PFD-CDS algorithm finds higher levels of commonality than the PFD-SSE algorithm. PFD-CDS uses the complete design space to explore possible solution. It is not limited by the solution spaces. Thus it can push the design variable values to the limits of the good design domain. By doing this PFD-CDS can find an optimized solution with similar design variable values as the PFD-SSE solution but higher commonality. This decreases the cost.

Nevertheless, both algorithms tend to a high commonality for both housing and reel and a rather individualized spring design.

Figure 7.26 shows the optimized design variables for product 25m according to PFD-CDS. The optimized DV values are at the limit lines of the good design domain. In contrast, figure 7.27 shows the optimized solution according to PFD-SSE. The optimized design variables do not lie at the very limits. This leads to both increased costs and increased robustness, as the distance to a bad solution is larger.

The calculation time of PFD-SSE (12s) was 275 times faster than PFD-CDS (55min). The time of PFD-SSE does not include the definition of the solution spaces.



Product		15	20	25	30	35
w_h	m	0.261				
d_h	m	0.517				
w_r	m	0.185				
d_r	m	0.235				
h_r	m	0.126				
h_s	m	7.26E-04	6.50E-04	7.80E-04	7.21E-04	
w_s	m	0.023	0.022	0.020	0.029	
l_s	m	7.10	12.21	12.62	14.99	
$d_{s,o}$	m	0.339	0.300	0.341	0.377	
$d_{s,i}$	m	0.025	0.026	0.026	0.025	

Figure 7.25: Optimized assignment scheme and design variable values for a product family of water hose boxes using PFD-CDS. Top: total cost of the product family over number of components variants. Middle: optimized commonality with components. Bottom: Optimized design variable values of the component variants.

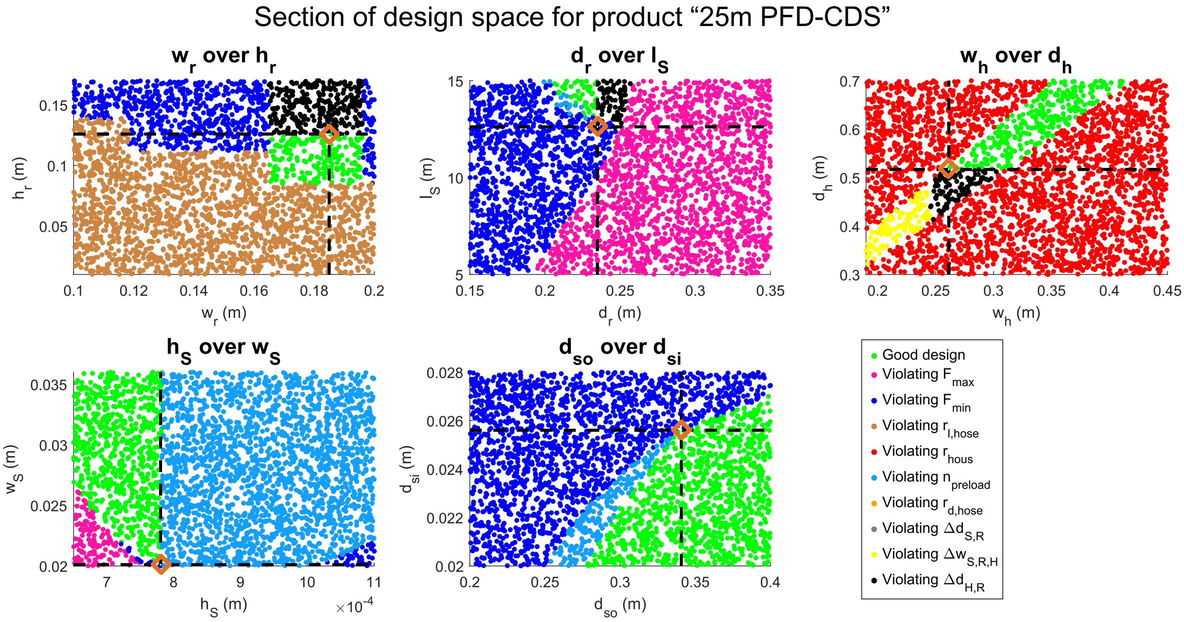


Figure 7.26: Sections of the design space at the resulting design of PFD-CDS for the product variant 25m. Dashed lines and the orange diamond indicate optimized design variable values.

	Commonality scenarios		
	low	middle	high
Complexity costs	low	middle	high
Discount rate	low	middle	high

Table 7.3: Definition of commonality scenarios

Three scenarios and three strategies

The cost parameters determine the objective function. They have an influence on the optimization results. Table 7.3 defines the commonality scenarios used for the following analysis. They are based on the cost model (see section 7.2.5).

The commonality scenarios define how strong the cost model supports commonality. High complexity costs penalize the introduction of new component variants. High discount rates benefit high volumes and high volumes can be created by using common component variants. Thus, the commonality scenario ‘high’ supports high commonality. The corresponding applies for the other commonality scenarios.

The solution spaces define where the optimization algorithm will search for optimized DV values and the assignment scheme. Thus, the choice of the solution spaces has an influence on the result of the optimization process. It also provides an interface for the user, where they can amend the optimization with implicit expert knowledge.

For the following analysis three strategies are defined: The solution spaces from section 7.2.6 are called PFD-SSE 1. PFD-SSE 2 describes a strategy where the solution spaces are moved to low DV values. This is done by, first, finding a point-based solution in the very bottom-left corner of each DV. Then, the upper bounds of each DV are expanded until they violate a requirement. PFD-CDS is the

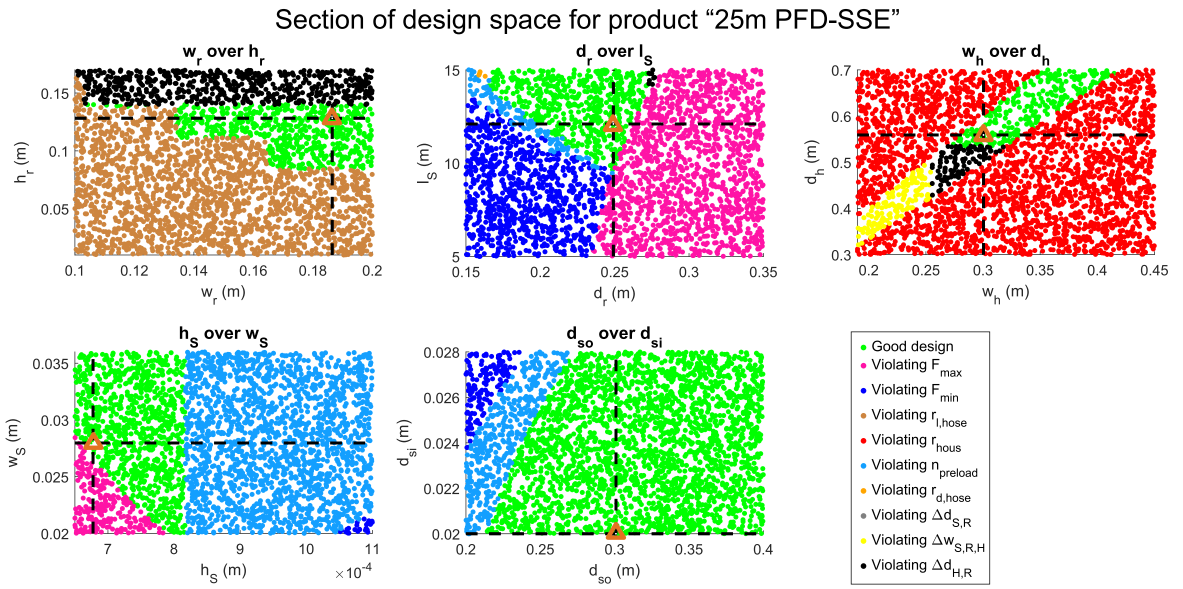


Figure 7.27: Sections of the design space at the resulting design of PFD-SSE 1 for the product variant 25m. Dashed lines and the orange triangle indicate optimized design variable values.

third strategy and is the same as described in section 7.2.6.

Figure 7.28 shows the results of the analysis with the commonality scenarios and the strategies described above. The commonality scenarios in the columns describe how favorable the cost parameters are for commonality. The rows indicate the strategy used to identify the optimized solution. The bold black dots indicate the optimized solution.

The influence of the commonality scenario can be observed in all strategies. The minima curve of cost-optimized assignment scheme changes its shape. The more favorable the commonality scenario (low - middle - high), the more costly solutions become with high numbers of component variants. The right side of the curve moves upwards and the left side downwards. Also the optimized solution changes in the PFD-SSE strategies: The more favorable the commonality scenario, the more the optimized solution (bold black dot) moves to the left, i.e., to solutions with less component variants, i.e., a higher commonality.

The choice of the solution spaces has a major impact on the considered assignment schemes. The solution spaces in strategy PFD-SSE 1 are bigger than in PFD-SSE 2. Therefore, more combinations to share components are possible. PFD-SSE 2 moves the solution spaces to small DV values. The solutions spaces also become smaller. Thus, only five assignment schemes are possible (including the pattern with individual component variants for each product variant). Higher commonalities than 13 component variants are not possible with this choice of solution spaces. The optimized solution gained from the PFD-CDS strategy stays the same for all scenarios. Nevertheless, the curve changes it shape which is due to the influence of the commonality scenario.

Table 7.4 compares the total product family costs and the mean computation times of the different commonality scenarios and strategies relatively with each other. The lowest values for each scenario (column) is per definition 100%. As expected, the PFD-CDS strategy yields the lowest values for all scenarios, as it searches the complete design space and pushes the DV to its limits (see figure 7.26). The table also reveals that the choice of the solution spaces by the user has an influence on the result.

In this example, the PFD-SSE 1 strategy (large solution spaces) leads to lower costs in the scenario

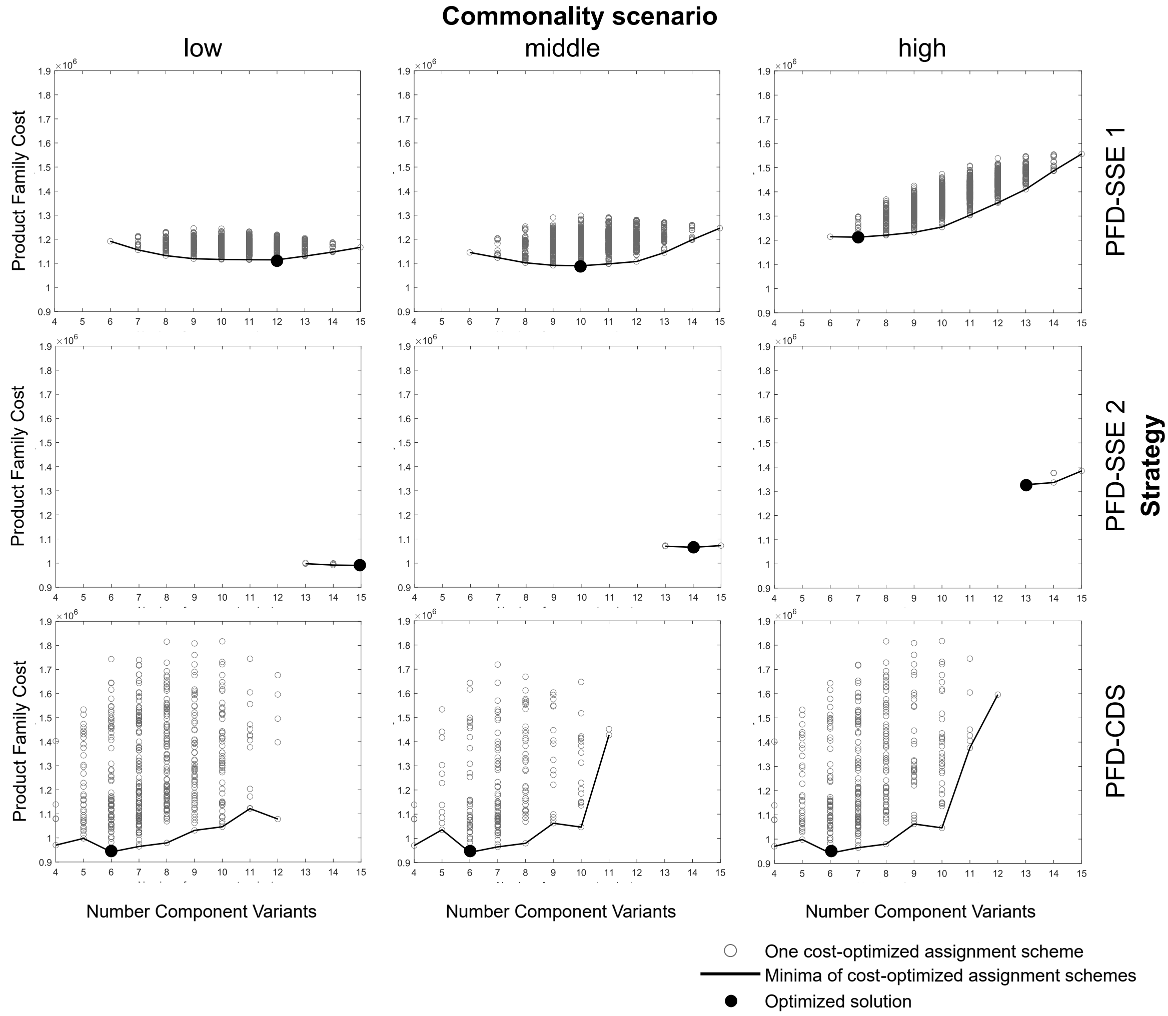


Figure 7.28: Analysis of the influence of commonality scenario and strategy on the optimized solution. Columns: commonality scenario indicating the influence of the cost parameters on the commonality. Columns: strategies to identify the optimized solution.

Commonality scenarios				
low	middle	high	Strategy	Mean computation time
118%	116%	129%	PFD-SSE 1	300%
105%	113%	141%	PFD-SSE 2	100%
100%	100%	100%	PFD-CDS	37 000%

Table 7.4: Relative comparison of total product family costs C_{PF} and the mean calculation time according to the commonality scenarios and strategies from figure 7.28. The lowest value within one column is defined as 100%

which favors commonality ('high') than the PFD-SSE 2 strategy (small DV values). For the scenario which supports low commonality the PFD-SSE 2 strategy yields lower costs.

In the 'low' commonality scenario the PFD-SSE 2 strategy and the PFD-CDS strategy lead to costs which differ by 5%, although the optimized assignment schemes are very different: PFD-SSE 2 leads to 15 component variants, whereas PFD-CDS leads to six component variants. Thus, completely different assignment schemes lead to very similar costs. The algorithm may have difficulty finding a direction of improvement or it may get caught in a local optimum. This illustrates the difficulty of this mixed integer non-linear problem statement that optimization algorithms must address.

The calculation times for the PFD-CDS approaches are significantly higher than for the PFD-SSE approaches (here: 123 - 370 times). This is consistent with previous findings. Additionally, PFD-SSE 2 was three times faster than PFD-SEE 1. Both approaches used exhaustive search to identify the cost-optimized product family. However, the solution spaces in PFD-SSE 2 are smaller and overlap less. Therefore, there are less assignment schemes the algorithm needs to optimize. Thus, the calculation time decreases.

8 Discussion

This chapter discusses the developed approach from chapter 6 against the goal and the requirements from chapter 4 using the results and experiences from the evaluation cases from chapter 7. Table 4.1 shows the requirements of the approach. The following headlines represent the categories: functional requirements (FRs) and application requirements (ARs).

8.1 Discussion of Functional Requirements

FR 1: Optimizes class III problems with generalized commonality

The developed approach optimizes both the assignment scheme and the DV values. It has no limitations on the commonality. It searches for the optimized solution among all assignment schemes as can be seen in the result figures of the optimized solutions, such as figure 6.13, 7.11, 7.24 or 7.25. The optimized assignment schemes are indicated by dots.

FR 2: Optimizes components with more than one design variable

The user can assign different DVs to one component. The approach then ensures that all design variables of that component have the same DV values when they are shared. Otherwise the component cannot be shared. Figure 7.11 shows the optimized solution of the electric vehicle example. Length and diameter describe the component motor. Same applies for the water hose example: two design variables describe the housing, three the reel and five the spring (see figure 7.24).

FR 3: Applicable during the early phases of the development process

The evaluation case of the electric vehicle design (see section 7.1 demonstrates how the developed approach can be used during the early phases of a development process.

This evaluation case is characterized by high levels of uncertainty. As typical in early phases, there is little knowledge of the final design but at the same time there are vast possibilities to influence the final design. For this application, especially the PFD-SSE algorithm is suitable.

It can not only provide a cost-optimized product family design, but also permissible intervals for other components or systems, which have an influence on the overall design but are not subject the optimization itself. The track width and the length of the wheelbase serve as an example. Even though the final design is not known at that point, the designers can define intervals in which a possible solution will achieve the required performance. This allows for design for commonality at an early stage of the product development process.

A designer can choose the intervals of the solution space manually and integrate implicit knowledge about the design. They can also define different intervals for the design variables and, by doing this, they assign different levels of flexibility: larger intervals lead to higher flexibility. This helps especially for very uncertain variables or variables which are beyond the designers' influence, such as the front surface or the drag coefficient of the electric vehicle.

Also the available data and models may change during the development process as there may be new insights. The model database provides the framework to exchange certain modules with new ones. This does not influence the rest of the models. It can also be extended with new models, which can describe components or sub systems in more detailed manner. For that, the new modular model has the former design variables of the system as an output. For instance, a model could describe the aerodynamic behaviour of a vehicle. This model could be connected to an existing model by the drag coefficient and the surface. Those would be the outputs of the modular model. The inputs could be geometrical attributes of the vehicle's surface. This allows for a vivid database which grows with the tasks and the knowledge gained during the development process.

FR 4: Includes requirements of the product variants

The first step of the developed approach is the collection of requirements (see chapter 6). Requirements distinguish the product variants from each other. The requirements are key for all following steps. The modelling of the system dependencies (step 2) starts with the collection of the quantities of interest, which are derived from the requirements. Solution spaces use requirements to divide good designs from bad designs. Both optimization algorithms ensure that all requirements are fulfilled.

FR 5: Supports resolving conflicts of goals

Conflicting goals can appear along different conflict lines: among technical goals, among economical goals, and between technical and economical goals. The developed approach includes both: technical and economical aspects, as well as their interaction.

Solution spaces visualize technical conflicting goals. The 2D projections and the interactive tool help to define the limits of the solution space such that all technical requirements are met and technical conflicts are solved.

From an economical perspective the conflict arises from the question: What is the cost optimal degree of standardization? Subsequent questions arise: How does the increase of production volume due to standardization reduce the costs per unit? How does over dimensioning increase costs? Where is the optimum? The result figure with the total product family cost over the number of component variants (e.g., figure 7.25) provides an solution for these questions. The optimization algorithm alters all cost relevant variables to find the optimized solution. The optimum differs from case to case and from scenario to scenario. This figure helps to identify the optimized solution.

The interaction between technical and economical perspective is handled differently in PFD-SSE and PFD-CDS. PFD-SSE defines solution spaces first and then searches for the best solution within these solution spaces, i.e., the possible degrees of standardization from a technical perspective. Within these limits the optimization algorithm considers technical and economical interaction.

However, PFD-SSE is limited to the choice of solution spaces. Different choices of solution spaces can lead to different results. Figure 7.28 in the water hose box case (see section 7.2.7) shows the influence of the choice of the solution space on the result. Solutions, which lie outside the box-shaped solution spaces, are not considered.

PFD-CDS does not have this limitation. It searches the complete design space to find the best fitting technical solution to minimize the product family cost. PFD-CDS can be used to set the benchmark to compare different strategies. In the analysis of the water hose box design (see table 7.4), PFD-CDS yielded lowest cost in all cost scenarios. Users can evaluate how costly a deviation from that solution will be.

However, also PFD-CDS may have difficulties to find the optimal solution due to mixed integer non linear nature of the problem statement. These complex optimization problems are challenging. The design influences the design dependent costs. At the same time the design determines the assignment scheme, which determines the production volumes of the different component variants. The production volumes of the component variants determine the effects of economies of scale, which have an influence on the design related costs as well. Completely different assignment schemes can lead to similar objective function values. This makes it difficult for optimization algorithms to find the optimal solution. So far, it is not possible to ensure that the global optimum of the optimization problem has been found.

As in any other model-based approach, the results highly depend on the quality of the models which are used. PFD-SSE provides a framework to find an optimized solution, which is not on the limit lines of the good design space. This can reduce the dependency of the result on the model accuracy.

FR 6: Allows to incorporate different disciplines

The approach connects the economical with the technical perspective. The objective function considers the economical perspective (see step 4, section 6.6). The model database incorporates the technical perspective (see step 3, section 6.5).

The technical perspective again can include different disciplines. The model database (step 3) based on the attribute dependency graphs (step 2) is a framework to incorporate different disciplines. Each modular model is a standalone function. It can be evaluated independently from the others. Each discipline which is involved in designing the product can provide its knowledge by models. They can put their domain specific expertise in their models. This can also increase the confidence in the merged system model. The motor model in the electric vehicle example (see section 7.2.4) was provided by an engineer specialized on the motor design. He could integrate his specific knowledge into this modular model. As part of the database it was then integrated in the system model.

Same can be done for any other discipline. However, the interfaces need to be defined. This requires an overview of the involved systems and the relevant attributes.

A major drawback of this approach is that the attributes and the dependencies need to be quantified. Attributes which are not quantifiable cannot be considered directly.

Modelling system dependencies and behaviour (step 2 & 3) requires the highest effort within the approach. It is often an iterative process which requires updating. The database with its templates and its structure (see section 6.5) can help to organize this process. The main effort of modelling however remains. It can be reduced when models are reused.

FR 7: Provides transparency during complex system design

Complexity arises from many different dependencies. The developed approach uses the following methods to handle complexity and increase transparency: attribute dependency graphs (step 2); the model database (step 3); visualizations of solution spaces, and the curve of optimized assignment schemes (step 5).

The clear cause-and-effect structure of ADGs helps the user to understand the cause-and-effect chains. In the water hose box example (see figure 7.18), the width of the reel determines together with the diameter of the water hose the number of horizontal layers. The number of vertical layers is defined by the number of horizontal layers, but also by the hose length, the hose diameter, and the diameter of the reel. The number of vertical layers determines together with the diameters of the reel and the hose the lever on which the torque provided by the spring applies. The lever and the torque determine a certain insertion force. If somebody starts designing the reels according to the required hose lengths, they would get certain values for the reel diameter, width and height. Then they can design the spring according to the insertion force. In the end they might come up with a spring which is too big for the reel. Then they need to re-design the reel so that the spring fits. This results in unintended iterations. The clear cause-and-effect structure of ADGs can avoid those iterations and provide transparency about the dependencies.

ADGs prevent circular dependencies. The modular model database uses this property. The lack of circular dependencies provides an unique flow of information. Modular models can be merged together automatically. At the same time modular models divide the complex system into manageable parts without loosing information. These parts can be modeled and validated independently. The algorithm groups them together in the right order (see section 6.5). Each model has a clear in- and output structure. These modular models with defined interfaces and structures increase transparency

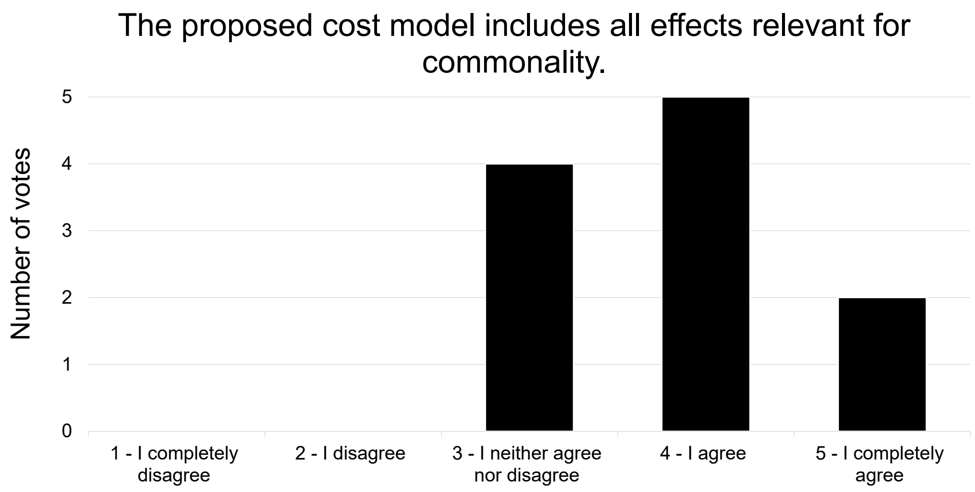


Figure 8.1: Eleven consultants evaluated the proposed cost model (objective function) on a five-point Likert-scale .

compared to a single, monolithic system model.

Solution spaces help to better understand the dependencies of a complex system - even if an engineer is familiar with the system. A person cannot imagine the influence of more than three attributes on the system behaviour simultaneously, as one of the engineers from the electric vehicle evaluation case said. Solution spaces helped him to understand the system better. Figure 7.14 also shows how different cycles influence the design. These diagrams visualize complex dependencies in a simple way without simplifying the system itself)as they are projections of the entire design space).

The diagrams showing the cost-optimized assignment schemes, the curve of local minima, and the optimized DV values provide information on the optimized solution. For example, the optimized solution of the water hose box family according to PFD-SSE (see figure 7.24) shows how the cost changes with lower or higher number of component variants. This provides information on the surrounding of the optimized solution, such as, how steep the increases in cost are when going to less or higher commonality. This also helps to better understand the solution of the complex design problem compared to a single optimized point only.

FR 8: The objective function incorporates relevant effects of standardization.

The objective functions of the evaluation cases presented above are based on the cost model described in section 6.6. To evaluate the elements of the cost model, the electric vehicle use case (see section 7.1) was presented to a consulting firm specialized on variant management and product family design. After the presentation the consultants evaluated the approach in a questionnaire.

Figure 8.1 shows the results of the evaluation. Eleven consultants evaluated the statement "The proposed cost model includes all effects relevant for commonality" on a five-point Likert-scale. None of them disagreed. Seven out of eleven agreed (five) or strongly agreed (two) to the statement.

In the evaluation cases, it was possible to collect the data which the cost model requires. But collecting this data is not trivial, as also one of the consultants mentions. Especially complexity cost are hard to measure. They are often subject to uncertainty.

The cost model can have a strong influence on the results of the optimization (see figures 7.12 and 7.28). This makes an analysis of the cost model necessary. The PFD-SSE approach, with its short calculation times, can quickly evaluate the influence of different cost models by changing the cost parameters while keeping the technical solution constant.

FR 9: Provides visualizations for decision support in product family design.

The approach provides visualizations for the different steps. Attribute dependency graphs (see figure 7.18 and 7.18) visualize the dependency structure of the problem. They support a common understanding which lays the basis for the following visualizations.

The figure, which summarizes the results of the optimization (see figure 7.11 and 7.24), condenses all the data and knowledge from the steps before. It consists of three parts:

The diagram on the top gives an overview of the total product family cost with respect to the number of components used. The user can see the neighbourhood of the optimized solution. As there may be other influences on the decision than only cost, such as, marketing strategies, the user can see how other optimized assignment schemes and designs behave compared to the optimized solution and how costly a deviation from that optimum might be.

The part in the middle illustrates the cost-optimized assignment scheme with icons. It helps the user to directly see how the optimum looks like and which product variants share which components. It is easier to comprehend than the plain numerical values.

The part at the bottom of the figure shows the concrete numerical values of the optimized solution. It also includes the derived solution spaces. This helps to check the plausibility of the results and also helps to explore the possibility to change the design within the limits of the solution spaces. Overlapping solution spaces indicate potential for commonality. These diagrams contain all the technical information in a condensed manner.

Another useful information is the maximized common solution space of the optimized solution (see figure 7.13). It shows the maximized flexibility to achieve the assignment scheme, which includes the optimized solution. The optimized solution is depicted by a triangle in the solution space. The limits of the solution space show the range in which the design variables can be varied such that the optimized assignment scheme can be still achieved for the complete product family. It maximizes the freedom of design changes. But design changes usually come at the expense of cost optimality.

If the user changes the values of the optimized design variables, the optimized assignment scheme will remain, but it is only cost optimized for DV values marked with the triangle. A change will lead to higher cost values. Another drawback is that the user does not know how much a change in DV values will worsen the overall cost.

FR 10: Allows for robust design of product families.

This section first discusses technical and then economical robustness.

In the use cases of this thesis, optimality and robustness are two sides of the same coin, i.e., low cost and robustness with respect to functionality are in conflict. ‘Classical’ optimization algorithms, represented by the PFD-CDS, typically search for the optimal solution. In engineering applications, the results mostly lie at the limits of the good design space (see figure 7.26). This makes the solutions very vulnerable to changes. Changes can appear due to changing requirements or updated models.

PFD-SSE can increase robustness and flexibility, if the solution spaces are big enough. Usually, this will lead to higher overall costs. But the user can directly influence the optimization process by changing those solution spaces. They can decide consciously for and against certain designs while ensuring the overall system performance by fulfilling all requirements. They can move the solution spaces ’away’ from certain limits or can use the maximized solution space for the optimized assignment scheme (see figure 7.13) to change the design to more robust solutions.

The developed approach provides both: a ‘classical’ algorithm which searches the optimum in the complete design spaces (PFD-CDS); and an algorithm which incorporates the user’s implicit knowledge in the form of solution spaces (PFD-SSE). A user can choose to run both approaches and compare the solutions and quantify the loss of optimality according to the objective function.

Economical robustness in this context describes a low sensitivity against changes of the cost model, such as changes in complexity costs or production volumes. The presence of such a robust solution is highly dependent on problem at hand. The approach can identify such a robust solution by a scenario analysis of the cost model. Different values of the uncertain parameters describe the scenarios. Each scenario is optimized and the results are compared (see figure 7.12 or 7.28). The user can see how the scenarios influence the optimized assignment scheme and if this is the case, how the DV values and the patterns will change. The approach was able to find an economically robust solution for the electric vehicle design (see figure 7.12) and the water hose box design (see figure 7.28). Especially, PFD-SSE is suitable for this kind of analysis due to its fast calculation times.

The evaluation of economical and technical robustness is rather qualitative. No measures of robustness were applied.

8.2 Discussion of Application Requirements

AR 1: Guides and supports the user during application.

The five steps of the procedure model guide the user through the application of the approach (see figure 6.1). They build the modular models according to the templates and examples (see section 6.5). They can validate the modular models independently from each other.

An algorithm automatically combines those modular models according to their ADGs. It builds the system model and generates the ADG of the system. This automatically generated ADG can be used to control the dependency structure of the system. The algorithm generates all necessary files for the following steps. The user only has to insert the requirements and the cost model. The cost model is also generated automatically as a dummy function with all relevant interface variables. The user then can introduce an own cost model or adapt the proposed one.

Both optimization algorithms PFD-SSE and PFD-CDS rely on the same data and models. Thus, users can easily switch between both algorithms. The tool guides the user during application. They can change the solution spaces in the PFD-SSE approach interactively by selecting and moving the limit lines of the box-shaped solution spaces with the cursor. They can generate solution spaces for different product variants. Later they can select the product variants they want to optimize, assign the design variables to components, define parameters, add production volumes and start the optimization. Same applies for PFD-CDS except that the user does not have to define the solution spaces of the different product variants.

The steps follow the procedure of the approach and support the user by templates, examples and automation. However, especially the model database is very restrictive. The user has to stick exactly to the templates. Otherwise the code will not work. It does not contain intelligent algorithms to support the user or detect errors. DVs need to be named the same, if they should be connected along different models.

Some of the diagrams are generated automatically, such as the diagram 'Product family costs vs number of component variants' (see figure 7.11, top) or the solution spaces (see figure 7.9). This supports the analysis of the results, but they still need to be edited for more sophisticated figures.

AR 2: Applicable independently of problem and user.

Different students applied the approach. This thesis presents the electric vehicle design and water hose box design cases as evaluation cases. The students show that the developed approach can be applied by different persons in different applications.

To evaluate the applicability, the electric vehicle use case (see section 7.1) was presented to a

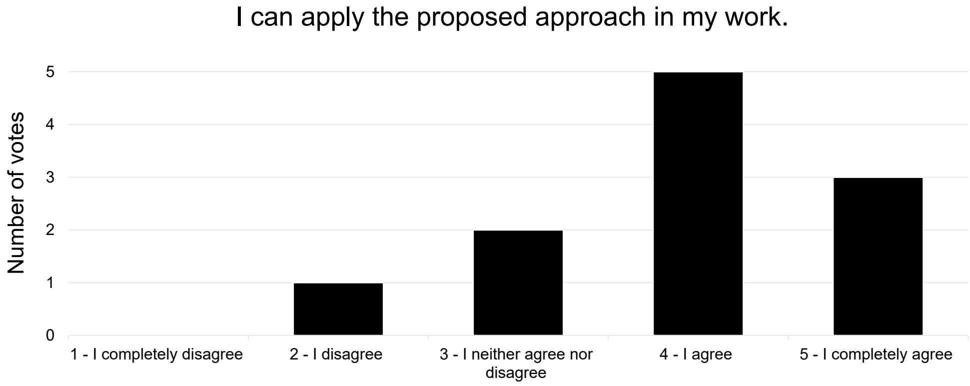


Figure 8.2: Eleven consultants evaluated the applicability of the developed approach in their work on a five-point Likert-scale .

consulting firm specialized on variant management and product family design. After the presentation the consultants evaluated the approach in a questionnaire. Figure 8.2 shows how the consultants rate the applicability of the developed approach in their work. Eight out of eleven agree (five) or strongly agree (three) that they can apply this approach in their work. One consultant disagrees. One comment says that a product cannot always described well with a certain set of attributes.

The empirical studies revealed that the interviewed consider the model database as highly applicable in their companies (see figure 3.6). To increase the applicability of the product family approach, the model database was included in the developed approach. Due to its generic input output structure, its application is not limited. The evaluation cases have shown that it can be applied in product family design. The interviewed see a high applicability in their companies. Thus, there is a high potential of applying this method. Especially product family design can create synergies for the model database. Product family design tries to re-use components in different products. When components are be re-used, also there models can be re-used. This can reduce the modelling effort on the long term.

On the short term, however, the modelling effort and the prerequisites are high. In contrast to qualitative approaches, the developed approach needs models to describe the system behaviour and the cost structure of the problem at hand. It has a list of prerequisites (see table 6.2) which need to be fulfilled. Most of them require a quantification of attributes, requirements or models. This can prevent or make the application of the approach more complicated in some cases - especially, when it comes to user experience. In contrast to performance criteria, user experience is hard to measure and constrain by quantitative requirements.

AR 3: Faster than existing approaches.

PFD-CDS is a representative of state of the art algorithms which can be applied to the evaluation cases at hand. Chapter 2 explains and section 4.2 summarizes the challenges for these kind of optimization algorithms.

PFD-SSE circumvents most of the challenges by using solution spaces. The user, supported by algorithms, can define box-shaped solution spaces. Box-shaped solution spaces decouple the design variables from each other. Furthermore, they contain only good designs, i.e., that constrains on the system behaviour do not need to be evaluated again. Both decreases the calculation time tremendously. The optimization of the water hose box example took 55min for the PFD-CDS algorithm and 12s for the PFD-SSE algorithm. This is 0.36% of the time of the PFD-CDS algorithm.

This decrease of calculation time comes along with a loss of solution space. Box-shaped solution spaces only use parts of the complete solution space. Thus the global optimum may not be found.

8.3 Contribution of this Thesis

This thesis contributes to research area of product family design with a *fast calculating approach* to solve a class III problem. This means it optimizes assignment scheme and design variables simultaneously. This thesis introduces two algorithms: one is based on state of the art algorithms (PFD-CDS), the other is based on solution spaces (PFD-SSE). PFD-CDS combines elements of existing approaches and represents the state of the art. PFD-SSE solves the class III problem faster than existing approaches.

This thesis also contributes to *robust* product family design. It provides an approach to optimize product family robustly. Solution spaces also provide flexibility during product family design.

The thesis also provides *current insights on industry related challenges* of product development, which lay the basis for the developed approach.

Many approaches focus on the optimization algorithms. This thesis presents the approach of product family design in five steps, including two optimization algorithms. It does not only focus on the algorithms but also on the *process to structure and systematically handle the product family design problem*. The evaluation cases show that these five steps are plausible and can guide users through the complex process of designing product families.

9 Summary and Outlook

9.1 Summary

Individualization of customer demands increases the variety of products companies need to offer on the market. This can lead to a growing internal variety. **High internal variety** causes cost along the development process and beyond, ranging from additional design and change efforts, storage and production costs, to documentation and supply of spare parts. At the same time, the complexity of the products increases. Both effects, variety and complexity, make it difficult for companies to identify designs which fulfill all requirements and are economical at the same time. Furthermore, it is necessary to design not only one product but the whole product family simultaneously to identify synergies. This makes the problem even more complex. Product family design is a major lever for companies to manage increasing variety and complexity and to reduce cost.

This thesis analyzes the challenges from both academia and industry to identify requirements for the development of a new approach. Literature distinguishes between modular and scale-based product family design. Modular product family design focuses on dividing the product architecture into manageable modules. This thesis considers design as the assignment of values to attributes a designer can control (design variables). Thus, the focus lies on **scale-based product family design**.

Scale-based product family design deals with two main problems: the optimal assignment of components to products (assignment scheme) and the optimal design of those components (assigning values to design variables). Some approaches reduce complexity by solving only one problem and assuming the other as given. Solving the **joint problem** (assignment scheme and component design) is a mixed integer non linear problem, as the assignment of components to products is an integer problem, the component design itself is usually a non linear problem. This makes the optimization challenging. Many approaches use two level optimizations to deal with the different nature of the problems. Non-gradient heuristics, such as genetic algorithms or particle swarm optimization, are widely used. However, the main drawback are **long calculation times**.

From an industry perspective the main challenges are: **managing requirements**, **handling complexity**, and **making decisions**. Requirements often lead to conflicting goals. The interdisciplinary nature of the systems makes it difficult to design a product holistically, especially during the early phases of the product development process. Companies are facing an increase in complexity of their products. Many design variables interact with each other and determine the system behaviour. Due to the high number of dependencies and design variables, the emerging system behaviour is difficult to predict. Cause and effect is hard to identify. This also influences decision making. Decisions are made with little knowledge of the consequences. Interviewees demand for data-based decisions. There is a lack of comprehensive visualizations which support decision making especially under uncertainty.

This thesis presents an approach to address the challenges from above. It consists of five steps: (1) collecting requirements, (2) modelling system dependencies, (3) modelling system behaviour, (4) modelling cost, and (5) optimizing the product family. Each step uses the outputs of the previous ones.

The requirements from **step (1)** guide the procedure. They build the basis for modelling the system dependencies in step (2). **Step (2)** uses so called attribute dependency graphs (ADGs). They are a hierarchical dependency model. Quantities of interests (QoIs) form the top level of an ADG, design variables (DVs) the bottom level. Requirements are the main source for the QoIs. QoIs determine how the product variants differ from each other. The flow of information is always bottom-up. As soon as a designer assigns values to the DVs, the attributes above are determined. This provides a clear cause and effect structure and prevents circular dependencies.

It is the basis for modelling the system behaviour in **step (3)**. The user divides the complete system model, the dependencies of which are documented as an ADG, into manageable pieces, so-called modular models. Each modular model is a standalone model, which can be determined and evaluated independently from each other. Engineers from different departments can build and validate their models independently. An algorithm combines the models automatically to the overall system model via defined interface attributes. This algorithm also provides a cost model as an objective function. The user can use and adapt the proposed cost model in **step (4)**. It includes cost relevant aspects of product family design, such as production volumes, effects of economies of scale, or complexity costs.

In **step (5)** the user can use two optimization algorithms: PFD-CDS and PFD-SSE. PFD-CDS uses the complete design space (CDS) to optimize the product family design (PFD). It is a two stage approach based on algorithms which are commonly used in literature: genetic algorithms for the optimization of the assignment scheme and particle swarm optimization for the design of the components. PFD-SSE uses solution space engineering (SSE) to optimize the product family design (PFD). A solution space consists of only good designs. A good design fulfills all requirements of the product (variant). A set of design variable values determines one specific design. Box-shaped solution spaces decouple the design variables from each other at the expense of a loss of good designs. The user determines the box-shaped solution spaces including implicit expert knowledge. 2D projections support the user in identifying the solution spaces. Overlapping solution spaces of different product variants indicate that components can be shared between those products from a technical perspective. An optimization algorithm determines then the cost optimized degree of commonality and the component design within those solution spaces.

In this thesis, the PFD-SSE algorithm could solve a joint problem **275 times faster** than the classical PFD-CDS algorithm. However, the overall costs of the optimized product family are usually higher using the PFD-SSE approach. Using box-shaped solution spaces instead of the complete design space speeds up the calculation but good designs are lost. This means that the optimization algorithm cannot consider them. In technical problem statements the optimized solution is often at the limit of one or more requirements. If the solution space does not include this optimum it cannot be found. At the same time, the optimized solutions of the complete design space are on the limits of good designs. So they do not provide robustness. Small changes in design variable values or in requirements can lead to bad solutions, i.e., designs which violate one or more requirements. At the same time, the flexibility of changing solutions is very limited with point-based solutions. They are optimized to the very limit leaving little or no possibility in adapting the solution to new or changing requirements. PFD-SSE offers the possibility for a **more robust and flexible** optimized design. Decoupled solution spaces provide intervals within designs can be changed independently without violating any requirements of the product family. Overlapping solution spaces directly visualize the user the intervals in which the component needs to be designed to be shared.

The approach provides **visualizations to support decisions** in product family design. A three-part figure summarizes the main results of the optimization (see figure 7.11 or figure 7.24). On the top a diagram of the overall product family cost over the overall number of used component variants shows the position of the optimum and its neighbourhood. It shows how costly a deviation from the optimum would be. The second part shows the optimized assignment scheme. The user can see which product variants share which component variants. The bottom part of the figure shows the concrete design variable values of the optimized component variants.

The solution space visualizations provide technical information about the product variants. They show which design variable values fulfill all requirements and where the optimized solution is located. A user can change the design of the products by moving the limits of the solution spaces and sees how the overall system behaviour changes. It visualizes conflicting technical goals and can make decisions more **transparent**.

The user can also optimize the product family with respect to different scenarios of the cost model. They can vary parameters, such as production volumes or complexity costs and analyze the influence on the result. This shows how robust an optimized product family design is with respect to changing economical parameters. This can support decision making in early phases of the development processes subject to uncertainty. In the use case of a startup's electric vehicle design a **robust optimum** was found.

The approach provides methods and tools along the **entire process of product family design**. It helps to manage complexity by clear visualizations of cause and effect and dividing the complex system into manageable parts. It provides visualization to support data-based decisions in product family design. It enables both technical and economical robust design. The user can choose between two optimization approaches to optimize a joint problem. They can compare the results to understand the trade-off between cost optimality and robustness of different solutions and choose the one which fits best the company's needs.

9.2 Outlook

This approach optimizes the internal variety for a given external variety. It does not **optimize the external variety**. This can be a further field of research. In some applications the question arises "How many product variants should we offer on the market to satisfy certain customers?" and "What should those variants look like?". In this case, not the requirements of all customers need to be satisfied but an economical optimum needs to be found. Therefore, the objective functions needs to be changed. It must be analyzed whether the approach is suitable for generalized problem statements of optimizing the external variety.

When talking about manufacturing of components the choice of the **manufacturing technology** could be included into the optimization problem. While some manufacturing technologies have little investment costs but relatively high costs per unit (e.g., additive manufacturing), other technologies have high investment costs and comparably low costs per unit (e.g., casting). The first would lead to rather individually designed component variants, the latter to rather commonly designed ones. The choice of the manufacturing technology can have a major influence on the design of the components. It can be interesting for companies not only to know how to design the product family, but also how to manufacture it cost-optimally. This would add another dimension of complexity to the joint optimization problem. The type of manufacturing technology would be another degree of freedom, influencing mainly the parameters of the cost model.

The presented approach is limited to scale-based product families. **Modular product families** can be beneficial for configurable products. Modules are often linked to certain customer-relevant functions. By adding or removing modules the functionality of a product can be configured. Modules are combined via interfaces, which often need to be standardized. An interesting question would be "How can the developed approach of this thesis can help to design optimized modules?". Especially solution spaces can be used to define independent intervals for interface variables, which can be used by designers of other modules. Thus, one module can be optimized while providing flexibility for other modules to be connected to the optimized module.

The presented approach requires quantifiable requirements, which are often performance or geometrical requirements. A field which was not treated so far is the integration of functional requirements of binary nature: a product can have it or not. Nevertheless, it must be designed depending on other components. An example may be a vibration reduction system for construction tools. Some tools may have it, others may not. Nevertheless, the whole family needs to be optimized. This topic is closely connected to modular/configurable product family design. Adding the possibility of optimizing also the configuration of certain components would widen the field of application. This would require a more flexible handling of **different product architectures**. So far, only product variants with the same product architecture can be optimized. When allowing configurations the product architecture may change.

Allowing **technology exchange** would require even more flexibility in product architectures. Similar to the choice of the manufacturing technology, some companies are interested in which technology to use in their product families. For instance, the function of switching a circuit with high current using a circuit of low current can be realized by a bipolar transistor or a mosfet. They differ in their technical specifications and the cost structure. Depending on the use case and the production volumes it may be reasonable to use one instead of the other. The approach would need to be able to choose between different technical concepts for the same functions/requirements in the context of product family design. It needs to be able to change the product architecture for all or some product variants. In this case, the structure of the model database could be beneficial. Different technical concepts could be modeled as different modular models which can then be combined according to defined interface attributes.

Technology exchange can also be thought the other way round: "How can components be not only shared among product variants of a product family, but among **different product families**?" In this case the component needs to be designed in different products with different requirements. Especially challenging is the fact that each product family has different system models. In this case, the solution space approach could help. When defining solution spaces for each product of the different product families, the solution spaces for the component which should be shared can be considered for the optimization. The solution spaces for the other design variables constrain the design of the other components, such that the requirements of each product family is fulfilled. This is similar to the electric vehicle design, where some attributes were used for the optimization of the product family, while other constrain yet unknown attributes, such as the drag coefficient.

Software plays a major role in product family design. It is a cost efficient way to make products distinguishable for customers. It has hardly any design related costs, but mainly complexity costs. Co-designing software and hardware of a product family would enable the possibility to identify synergies and save hardware costs. This also provides more flexibility as software can be changed later on. If this is quantified, a user can use the solution space to explore the influence of software on the system behaviour. This can be especially interesting for controlled systems. Co-designing the plant and the control can lead to cost reduction across the entire product family.

Another feature which was not used during the evaluation examples is that only some design variables of a component should be common while others can differ. This could be interesting for the spring in the water hose box example. While width and height of the spring steel could be the same, the length can be different. Then the company could buy one coil of spring steel for all product variants and just cut different lengths out of it. This would allow for a more **flexible component design** while still using effects of standardization. This could be done by defining the width and height as separate components and the length as another component. Then also cost model needs to be updated accordingly.

10 List of Figures

11 List of Tables

12 List of Student Projects

The following student projects were written in the context of this thesis. The author of this thesis supervised the student projects. They are listed in a chronological order.

Thoma, Dominik (2019). Numerische Optimierung der Komponentenvielfalt unter Verwendung der Lösungsraumanalyse (Master's thesis). Technical University of Munich, Munich.

Stulpe, Julia (2020). Herausforderungen in der Produktentwicklung und Anwendungspotential von Solution Space Engineering (Master's thesis). Technical University of Munich, Munich.

Rostan, Nicky (2020). Sequencing of Modular Models in Systems Design (Master's thesis). Technical University of Munich, Munich.

Steger, Hans (2020). Robuste Reglerauslegung für modulare Systeme mit parametrischen Unsicherheiten mittels Solution Space Engineering (Bachelor's thesis). Technical University of Munich, Munich.

Daschner, Christoph (2020). Produktfamilienauslegung geregelter mechanischer Systeme mit Solution Space Engineering (Bachelor's thesis). Technical University of Munich, Munich.

Le Bourgeois, Martin (2020). Systematic Evaluation and Improvement of the approach: Cost Optimization of Product Families using Solution Spaces (Master's thesis). Technical University of Munich, Munich.

Steger, Hans (2021). Numerische Lösungsraumoptimierung für kostenoptimierte Produktfamilien mittels Solution Space Engineering (Semester's thesis). Technical University of Munich, Munich.

Daschner, Christoph (2021). Produktfamilienauslegung von Batteriepacks mit Solution Space Engineering (Semester's thesis). Technical University of Munich, Munich.

Berger, Vincent (2021) Kostenoptimierte Produktfamilienauslegung einer Fahrzeugplattform mit Solution Space Engineering (Master's thesis). Technical University of Munich, Munich.

Stricker, Stella (2022). Product Family Design: Application to an Electrotechnical Use Case. (Master's thesis). Technical University of Munich, Munich.

13 References

Agard, B. (2004). Data-mining-based methodology for the design of product families. *International Journal of Production Research*, 42:2955–2969.

Albers, A. and Gausemeier, J. (2012). Von der fachdisziplinorientierten Produktentwicklung zur Vorausschauenden und Systemorientierten Produktentstehung. In Anderl, R., Eigner, M., Sendler, U., and Stark, R., editors, *Smart Engineering: Interdisziplinäre Produktentstehung*, pages 17–29. Springer Berlin Heidelberg, Berlin, Heidelberg.

Barathwaj, N., Raja, P., and Gokulraj, S. (2015). Optimization of assembly line balancing using genetic algorithm. *Journal of Central South University*, 22(10):3957–3969.

Baylis, K., Zhang, G., and McAdams, D. A. (2018). Product family platform selection using a Pareto front of maximum commonality and strategic modularity. *Research in Engineering Design*, 29(4):547–563.

Bell, E. T. (1934). Exponential Polynomials. *Annals of Mathematics*, 35(2):258–277.

Blessing, L. T. and Chakrabarti, A. (2009). *DRM, a Design Research Methodology*. Springer London, London.

Chen, C. and Wang, L. (2008a). Multiple-platform based product family design for mass customization using a modified genetic algorithm. *Journal of Intelligent Manufacturing*, 19(5):577–589.

Chen, C.-b. and Wang, L.-y. (2008b). A modified genetic algorithm for product family optimization with platform specified by information theoretical approach. *Journal of Shanghai Jiaotong University (Science)*, 13(3):304–311.

Chowdhury, S., Messac, A., and Khire, R. (2011). Comprehensive Product Platform Planning $(CPˆ3)$ Framework. *Journal of Mechanical Design*, 133:101004.

Chowdhury, S., Messac, A., and Khire, R. (2013). Investigating the commonality attributes for scaling product families using comprehensive product platform planning (CP3). *Structural and Multidisciplinary Optimization*, 48(6):1089–1107.

Cormier, P., van Horn, D., and Lewis, K. (2009). Investigating the use of (re) configurability to reduce product family cost and mitigate performance losses. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 1089–1100.

Dai, Z. and Scott, M. J. (2005). Effective Product Family Design Using Preference Aggregation. *Journal of Mechanical Design*, 128(4):659–667.

de Neufville, R. and Scholtes, S. (2011). *Flexibility in engineering design*. Engineering systems. MIT Press, Cambridge, Mass.

de Weck, O. L. (2006). Determining product platform extent. In *Product Platform and Product Family Design*, pages 241–301. Springer.

de Weck, O. L., Suh, E. S., and Chang, D. (2003). Product family and platform portfolio optimization. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 175–185.

Doraszelski, U. and Draganska, M. (2006). MARKET SEGMENTATION STRATEGIES OF MULTIPRODUCT FIRMS*. *The Journal of Industrial Economics*, 54(1):125–149.

Du, X., Jiao, J., and Tseng, M. M. (2001). Architecture of product family: fundamentals and methodology. *Concurrent Engineering*, 9(4):309–325.

Ehrlenspiel, K., Kiewert, A., Mörtl, M., and Lindemann, U. (2020). *Kostengünstig Entwickeln und Konstruieren: Kostenmanagement bei der integrierten Produktentwicklung*. Springer Vieweg, Berlin and Heidelberg, 8. auflage edition.

Eichstetter, M., Müller, S., and Zimmermann, M. (2015). Product Family Design With Solution Spaces. *Journal of Mechanical Design*, 137(12).

Eppinger, S. D. and Browning, T. R. (2012). *Design structure matrix methods and applications*.

Erens, F. and Verhulst, K. (1997). Architectures for product families. *Computers in Industry*, 33(2-3):165–178.

Ericsson, A. and Erixon, G. (1999). *Controlling design variants: modular product platforms*. Society of Manufacturing Engineers.

Farrell, R. S. and Simpson, T. W. (2003). Product platform design to improve commonality in custom products. *Journal of Intelligent Manufacturing*, 14(6):541–556.

Feitzinger, E. and Lee, H. L. (1997). Mass customization at Hewlett-Packard: the power of postponement. *Harvard business review*, 75:116–123.

Feldhusen, J. and Grote, K.-H. (2013). *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*. Springer Berlin Heidelberg, Berlin, Heidelberg, 8. aufl. 2013 edition.

Fellini, R., Kokkolaras, M., and Papalambros, P. Y. (2006). Quantitative platform selection in optimal design of product families, with application to automotive engine design. *Journal of Engineering Design*, 17(5):429–446.

Fujita, K. (2002). Product variety optimization under modular architecture. *Computer-Aided Design*, 34:953–965.

Fujita, K., Amaya, H., and Akai, R. (2013). Mathematical model for simultaneous design of module commonalization and supply chain configuration toward global product family. *Journal of Intelligent Manufacturing*, 24(5):991–1004.

Fujita, K. and Yoshida, H. (2004). Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes. *Concurrent Engineering*, 12(2):105–118.

Gao, F., Xiao, G., and Simpson, T. W. (2009). Module-scale-based product platform planning. *Research in Engineering Design*, 20(2):129.

Gebhardt, N., Kruse, M., Krause, D., and Lindemann, U. (2016). Gleichteile-, Modul-und Plattformstrategie. In *Handbuch Produktentwicklung*, pages 111–149. Hanser.

Gershenson, J. K., Prasad, G. J., and Zhang, Y. (2003). Product modularity: Definitions and benefits. *Journal of Engineering Design*, 14(3):295–313.

Gläser, J. and Laudel, G. (2010). *Experteninterviews und qualitative Inhaltsanalyse als Instrumente rekonstruierender Untersuchungen*. Lehrbuch. VS Verl. für Sozialwiss, Wiesbaden, 4. aufl. edition.

Graff, L., Harbrecht, H., and Zimmermann, M. (2016). On the computation of solution spaces in high dimensions. *Structural and Multidisciplinary Optimization*, 54(4):811–829.

Han, X., Li, R., Wang, J., Ding, G., and Qin, S. (2020). A systematic literature review of product platform design under uncertainty. *Journal of Engineering Design*, 31(5):266–296.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.

Holtta, K. M. M. and Salonen, M. P. (2003). Comparing three different modularity methods. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 533–541.

Jiao, J., Simpson, T. W., and Siddique, Z. (2007). Product family design and platform-based product development: a state-of-the-art review. *Journal of Intelligent Manufacturing*, 18(1):5–29.

Kazemzadeh, R. B., Behzadian, M., Aghdasi, M., and Albadvi, A. (2008). Integration of marketing research techniques into house of quality and product family design. *The International Journal of Advanced Manufacturing Technology*, 41(9):1019.

Kennedy, J. and Russell Eberhart (1995). Particle swarm optimization.

Khajavirad, A., Michalek, J., and Simpson, T. (2007). A Decomposed Genetic Algorithm for Solving the Joint Product Family Optimization Problem. In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Structures, Structural Dynamics, and Materials and Co-located Conferences. American Institute of Aeronautics and Astronautics.

Khajavirad, A., Michalek, J. J., and Simpson, T. W. (2009). An efficient decomposed multiobjective genetic algorithm for solving the joint product platform selection and product family design problem with generalized commonality. *Structural and Multidisciplinary Optimization*, 39(2):187–201.

Kipp, T. and Krause, D. (2009). Computer aided size range development-data mining vs. optimization. In *DS 58-4: Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 4, Product and Systems Design, Palo Alto, CA, USA, 24.-27.08. 2009*, pages 179–190.

Kirchner, E. (2020). *Werkzeuge und Methoden der Produktentwicklung: Von der Idee zum erfolgreichen Produkt*. Springer Vieweg, Berlin.

Kotler, P. and Armstrong, G. (2013). Principles of Marketing (16th Global Edition).

Krause, D. and Gebhardt, N. (2018). *Methodische Entwicklung modularer Produktfamilien*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kristianto, Y., Helo, P., and Jiao, R. J. (2013). Mass customization design of engineer-to-order products using Benders' decomposition and bi-level stochastic programming. *Journal of Intelligent Manufacturing*, 24(5):961–975.

Kumar, D., Chen, W., and Simpson, T. W. (2009). A market-driven approach to product family design. *International Journal of Production Research*, 47(1):71–104.

Li, L. and Huang, G. Q. (2009). Multiobjective evolutionary optimisation for adaptive product family design. *International Journal of Computer Integrated Manufacturing*, 22(4):299–314.

Li, S., Liu, Z., Wang, L., Chen, J., Ni, M., and Zhang, L. (2019). Full Constraint Parallel Optimization Algorithm for Structural Optimization of Product Family. In *2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, pages 109–112.

Li, Z., Feng, Y., Tan, J., and Wei, Z. (2008). A methodology to support product platform optimization using multi-objective evolutionary algorithms. *Transactions of the Institute of Measurement and Control*, 30(3-4):295–312.

Lindemann, U. (2016). *Handbuch Produktentwicklung*. Carl Hanser Verlag GmbH Co KG.

Liu, Z., Wong, Y. S., and Lee, K. S. (2011). A manufacturing-oriented approach for multi-platforming product family design with modified genetic algorithm. *Journal of Intelligent Manufacturing*, 22(6):891–907.

Liu Zhuo, Wong Yoke San, and Lee Kim Seng (2009). Modified GA-based optimizer for multi-objective product family design. In *2009 4th International Conference on Autonomous Robots and Agents*, pages 121–126.

Martin, M. V. and Ishii, K. (2002). Design for variety: developing standardized and modularized product platform architectures. *Research in Engineering Design*, 13(4):213–235.

Martins, J. R. R. A. and Lambe, A. B. (2013). Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075.

MATLAB (2020). *version 9.9.0.1495850 (R2020b)*. The MathWorks Inc., Natick, Massachusetts.

McGrath, M. E. (1995). *Product strategy for high-technology companies: how to achieve growth, competitive advantage, and increased profits*. Irwin Professional Pub.

Messac, A., Martinez, M. P., and Simpson, T. W. (2002). Effective Product Family Design Using Physical Programming. *Engineering Optimization*, 34(3):245–261.

Meyer, M. H. (1997). Revitalize your product lines through continuous platform renewal. *Research-Technology Management*, 40(2):17–28.

Meyer, M. H. and Lehnerd, A. P. (1997). *The Power of Product Platforms*. Simon and Schuster.

Meyer, M. H. and Utterback, J. M. (1992). The product family and the dynamics of core capability.

Misoch, S. (2019). *Qualitative interviews*. De Gruyter Oldenbourg.

Mitschke, M. and Wallentowitz, H. (2014). *Dynamik der Kraftfahrzeuge*. Springer Fachmedien Wiesbaden, Wiesbaden.

Moon, S. K., Park, K. J., and Simpson, T. W. (2014). Platform design variable identification for a product family using multi-objective particle swarm optimization. *Research in Engineering Design*, 25(2):95–108.

P. Tarasewich and S. K. Nair (2001). Designer-moderated product design. *IEEE Transactions on Engineering Management*, 48(2):175–188.

Papalambros, P. Y. and Wilde, D. J. (2017). Principles of Optimal Design: Modeling and Computation. *13164510*.

Pine, B. J. (1993). Making mass customization happen: Strategies for the new competitive realities. *Planning Review*, 21(5):23–24.

Ponn, J. and Lindemann, U. (2011). *Konzeptentwicklung und Gestaltung technischer Produkte: systematisch von Anforderungen zu Konzepten und Gestaltlösungen*. Springer-Verlag.

Reitan, J., Stormo, S., and Aasland, K. (2002). Redesigning Product Programs Focusing On Variety, Complexity and Commonality. In *DS 30: Proceedings of DESIGN 2002, the 7th International Design Conference, Dubrovnik*, pages 235–240.

Robertson, D. and Ulrich, K. (1998). Planning for product platforms. *Sloan management review*, 39(4):19.

Rötzer, S., Berger, V., and Zimmermann, M. (2022a). Cost Optimization of Product Families Using Solution Spaces: Application to Early-Stage Electric Vehicle Design. *Proceedings of the Design Society*, 2:583–592.

Rötzer, S., Le Bourgeois, M., Thoma, D., and Zimmermann, M. (2021). TWO-LEVEL OPTIMIZATION OF PRODUCT FAMILIES: APPLICATION TO A PRODUCT FAMILY OF

WATER HOSE BOXES. *Proceedings of the Design Society*, 1:3259–3268.

Rötzer, S., Rostan, N., Steger, H., Vogel-Heuser, B., and Zimmermann, M. (2020a). Sequencing of information in modular model-based systems design. In *22nd International Dependency and Structure Modelling Conference (DSM 2020).*

Rötzer, S., Schweigert-Recksiek, S., Thoma, D., and Zimmermann, M. (2022b). Attribute Dependency Graphs - Modeling Cause and Effect in Systems Design. *Design Science*, (in press).

Rötzer, S., Thoma, D., and Zimmermann, M. (2020b). Cost optimization of product families using solution spaces. In *Proceedings of the Design Society: DESIGN Conference*, volume 1, pages 1087–1094. Cambridge University Press.

Rupp, M. (1988). *Produkt-, Marktstrategien: Handbuch zur marktsicheren Produkt-und Sortimentsplanung in Klein-und Mittelunternehmungen der Investitionsgüterindustrie*. Verlag Industrielle Organisation.

Salvador, F. (2007). Toward a Product System Modularity Construct: Literature Review and Reconceptualization. *IEEE Transactions on Engineering Management*, 54(2):219–240.

Schuh, G., Lenders, M., and Arnoscht, J. (2009). Focussing product innovation and fostering economies of scale based on adaptive product platforms. *CIRP Annals*, 58(1):131–134.

Schuh, G. and Riesener, M. (2017). *Produktkomplexität managen: Strategien - Methoden - Tools*. Hanser eLibrary. Carl Hanser Verlag GmbH & Co. KG, München, 3., vollständig überarbeitete auflage edition.

Scott, M. J., Garcit' a Arenillas, Juan Carlos, Simpson, T. W., Valliyappan, S., and Allada, V. (2006). Towards a suite of problems for comparison of product platform design methods: A proposed classification. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 833–844.

Simpson, T. W. (2004). Product platform design and customization: Status and promise. *Ai Edam*, 18(1):3–20.

Simpson, T. W. and D'Souza, B. S. (2004). Assessing Variable Levels of Platform Commonality Within a Product Family Using a Multiobjective Genetic Algorithm. *Concurrent Engineering*, 12(2):119–129.

Simpson, T. W., Jiao, J., Siddique, Z., and Hölttä-Otto, K. (2014). *Advances in Product Family and Product Platform Design: Methods & Applications*. SpringerLink Bücher. Springer New York, New York, NY.

Simpson, T. W., Maier, J. R., and Mistree, F. (2001). Product platform design: method and application. *Research in Engineering Design*, 13(1):2–22.

Simpson, T. W., Siddique, Z., and Jiao, J. R. (2006). *Product Platform and Product Family Design: Methods and Applications*. SpringerLink Bücher. Springer US, New York, NY.

Song, H., Zhang, Y., Song, Y., Wang, Z., and Zhen, L. (2008). Product Platform Planning: an approach using Genetic Algorithm. In *2008 IEEE International Conference on Networking, Sensing and Control*, pages 1621–1625.

Stulpe, J. (2020). *Herausforderungen in der Produktentwicklung und Anwendungspotential von Solution Space Engineering: (Master's thesis)*. Technical University of Munich, Munich.

Suh, E. S., de Weck, O., Kim, I. Y., and Chang, D. (2007). Flexible platform component design under uncertainty. *Journal of Intelligent Manufacturing*, 18(1):115–126.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, 24(3):419–440.

Wang, W. (2011). Scalable platform design optimization using hybrid co-evolutionary algorithms. In *2011 IEEE 2nd International Conference on Software Engineering and Service Science*, pages 270–273. IEEE.

Wang, Z. and Ning, F. (2007). Study on Supply Chain Management for Product Family in Mass Customization. In *2007 IEEE International Conference on Automation and Logistics*, pages 1515–1519.

Wazed, M. A., Ahmed, S., and Nukman, Y. (2010). Commonality in manufacturing resources planning–issues and models: a review. *European Journal of Industrial Engineering*, 4(2):167–188.

Wei, W., Feng, Y., Tan, J., and Li, Z. (2009). Product platform two-stage quality optimization design based on multiobjective genetic algorithm. *Computers & Mathematics with Applications*, 57(11):1929–1937.

Wei, W., Ji, J., Wuest, T., and Tao, F. (2017a). Product family flexible design method based on dynamic requirements uncertainty analysis. *Procedia CIRP*, 60:332–337.

Wei, W., Pingyuan, W., Zhenyu, T., Jun, J., and Jin, C. (2017b). General Motor Family Flexibility Design Based on NICA-II. *Procedia Manufacturing*, 10:92–99.

Wei, W., Tian, Z., Peng, C., Liu, A., and Zhang, Z. (2019). Product family flexibility design method based on hybrid adaptive ant colony algorithm. *Soft Computing*, 23(20):10509–10520.

Wortmann, J. C., Muntslag, D. R., and Timmermans, P. J. M. (1996). *Customer-driven Manufacturing*. Springer Netherlands, Dordrecht.

Yu, T.-L., Yassine, A. A., and Goldberg, D. E. (2003). A Genetic Algorithm for Developing Modular Product Architectures. pages 515–524.

Zhuo, L., San, W. Y., and Seng, L. K. (2007). Towards effective multi-platforming design of product family using genetic algorithm. In *2007 IEEE International Conference on Automation Science and Engineering*, pages 300–305. IEEE.

Zimmermann, M., Königs, S., Niemeyer, C., Fender, J., Zeherbauer, C., Vitale, R., and Wahle, M. (2017). On the design of large systems subject to uncertainty. *Journal of Engineering Design*, 28(4):233–254.

Zimmermann, M. and von Hoessle, J. E. (2013). Computing solution spaces for robust design. *International Journal for Numerical Methods in Engineering*, 94(3):290–307.

A Transcribed Interviews

Nr. Interview	00	Unternehmen	X	Datum und Ort	22.05.2020, München	Interviewer	XXX
Probeinterview. Zoom Meeting, audio- und visuell (Freier Tag), keine Störungen und angenehme Atmosphäre. Dauer: 01:02:0							

[02:17] **Kannst du noch mal ganz kurz was zu deiner Person sagen und was deine Aufgabe im Unternehmen ist.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[04:30] **Wie lange arbeitest du ungefähr schon in deiner Position?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[05:01] **Ich nehme an, du warst in anderen Abteilungen bei XXXXXXXXXXX schon oder ist das deine erste?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

BEGINN HAUPTTEIL

[05:24] **Wenn du dir die Herausforderungen anschaut, fällt dir irgendwie noch ein relevantes Themenfeld ein, das hier fehlt? Irgendein Themenfeld, indem deiner Meinung nach auch viele Herausforderungen entstehen.**

Also grundsätzlich ist das Thema Komplexität. Das andere sind ja eigentlich spezifische Felder des bedeutet Anforderungen, Entscheidung, Variantenmanagement. Und ich finde Komplexität hängt über allen Themen drüber, weil die Komplexität, die hat man beim Variantenmanagement, beim Anforderungsmanagement, die hat man bei der Entscheidungsfindung.
Die Frage ist, ist es eine losgekoppelte Herausforderung? Es spielt irgendwie überall mit rein. Es gibt technische Komplexität, organisatorische Komplexität. Dementsprechenden fällt es da überall mit rein meiner Meinung nach.

[06:23] **Wenn ich da kurz einhacken darf. Das sehe ich auch so. In dem Fall ist eher die Komplexität der Produkte selber gemeint. Also die Produktstrukturen etc.**

Ok.
Was für mich in der frühen Phase noch eine zentrale Herausforderung ist, ist die Unschärfe, dass man nicht so genau weiß, was man selber will.
Da gibt es ja diese schönen Abbildungen, man legt am Anfang viel fest, aber eigentlich weiß man am wenigsten, was man möchte, weil man noch relativ vage ist. Und ich glaube, das ist so ein zentrales Spannungsfeld meiner Meinung nach, wie man das eigene Unwissen, wo man eigentlich hinmöchte, mit den Entscheidungen koppeln kann, die man treffen muss.

[07:23] **Also die Orientierung?**

Ja, genau das ist ein zentrales Thema.

[08:17] **Du kannst jetzt die Herausforderungen der Relevanz nach sortieren. Nach dieser Reihenfolge werden wir dann die Fragen durchgehen.**

1. Komplexität 2. Variantenmanagement 3. Entscheidungsfindung 4. Robustes Design 5. Orientierung 6. Anforderungsmanagement

---

KOMPLEXITÄT

[09:29] **Also hast du das Gefühl, dass die Komplexität der Produkte dich vor Herausforderungen stellt. Und wenn ja, welche?**

Also die Frage ist jetzt natürlich, stellt sie mich vor Herausforderung oder eigentlich auf die Organisation, weil man ist ja nur ein Teil davon. Ich glaube die Komplexität natürlich stellt mich persönlich vor bestimmte Herausforderungen. Aber gerade so große Unternehmen, wie zum Beispiel XXXXXXXXXXX, würde ich auf jeden Fall sagen, dass es da Herausforderung gibt.

Gerade wenn man sich jetzt vorstellt in diesem Kontext zum Beispiel. Man sieht es jetzt bei XXXXXXXXXXXX. Die XXXXXXXX haben einen immer größeren Softwareanteil. Das bedeutet, man hat einmal Sensorik, man hat gleichzeitig auch Software, die programmiert werden muss. Das muss zueinander passen, weil sonst nichts mehr funktioniert und diese Software Komplexität, die gab es in der Vergangenheit nicht. Und die ist schon eine zentrale Herausforderung, weil es nochmal eine neue Domäne ist, die dazukommt und die aber auch abgestimmt sein muss mit den anderen Sachen.
XXXXXXXXXXXX ist ein Beispiel wo man einmal die technischen Komponenten als Herausforderung hat, bestimmte Sensorik, und aber gleichzeitig auch die Logiken und auch das Verhalten XXXXXXXX programmieren und beschreiben muss, was sehr, sehr kompliziert ist und zusammenpassen muss.

[10:54] **Also das Zusammenspiel zwischen Software, Hardware und die bestehende Sensorik?** Ja genau. **Das wäre ja auch eine Ursache für diese Komplexität. Fällt dir noch eine andere Ursache ein?**

Die Anzahl der Bauteile bei den verschieben Branchen steigt.
Wenn man jetzt wieder auf den Vergleich XXXXX geht, da es vielmehr Regelung und vielmehr elektrische Komponenten drin, als es früher war, die alle miteinander vernetzt werden müssen. Es gibt viel mehr Komfort Funktionen von XXXXXXX bis irgendwie XXXXXXXXXXXXXXXXXXXXXXXXX. Das ist eine zentrale Herausforderung.

Dann denke ich die regulatorischen Anforderungen, die das schwieriger machen sie zu erfüllen. Gleichzeitig ist in vielen Sachen, glaube ich, eine Sättigung in der Technologie erreicht worden. Das bedeutet, man muss größere Anstrengung betreiben, um die Zielwerte auch noch zu erreichen. Dementsprechend wird die Technologie dann auch immer komplexer oder umfangreicher.

[12:20] **Und diese Ursachen - beziehungsweise ich will es mal als Kundenanforderungen bezeichnen, zum Beispiel diese XXXXXXXXXX. Dafür gibt es ja inzwischen auch einige Werkzeuge, wie Stakeholderanalysen und so weiter. Wenn du jetzt an die Komplexität denkst und den Ursachen, habt ihr da Methoden oder Werkzeuge, wie ihr mit der Komplexität umgeht?**

Ich bin jetzt nicht in der Produktentwicklung, deswegen fällt es mir da ein bisschen schwer, das genau zu sagen. Ich kenne DSM, Matrizen und so was.
Aber ich weiß ehrlich gesagt nicht im Detail, wie die Kollegen das in der Entwicklung machen. Aber ich weiß, dass zum Beispiel umfangreich Anforderungslisten aufgenommen werden, dass früh mit Anforderungen gestartet wird. Versucht wird zu Clustern, Abhängigkeiten auch zu erkennen. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Schnittstellen denke ich, werden definiert. Das ist das, was ich sagen würde, dass da gemacht wird, um das ein bisschen handhabbar zu machen und aber auch die unterschiedlichen Abteilungen daran arbeiten zu lassen.

[13:47] **Wie viele Personen arbeiten durchschnittlich an der Entwicklung eines Produkts? Beziehungsweise aus wie viel Abteilungen, kannst du da eine grobe Hausnummer geben?**

Also eine Abteilung ist ja erstmal von Unternehmensgrößen unterschiedliche. Eine Abteilung bei uns ist schon eher groß, da sind dann teilweise XX Leute drin. In anderen Unternehmen ist das wahrscheinlich kleiner, deswegen tue ich mich da ein bisschen schwer.
Alleine bei den Umfängen, wo ich dabei bin, schätze ich mal...
Die Entwicklung läuft bei uns zwischen Einkauf, Entwicklung und Produktion Hand in Hand und von jeder Seite kommen Stakeholder hinzu. Ich schätze, dass es für den jeweiligen Umfang dann schon in jeweiligen XX Abteilungen oder XX stattfindet.

Aber ja, es fällt mir schwer genau zu sagen, wie viele Mitarbeiter da jetzt genau daran arbeiten, weil natürlich die XXXXXX in so kleine Komponenten geschnitten werden, wo immer Leute daran arbeiten. Aber auf der nächsten Ebene sind es dann XXX Abteilungen, also XXX Äste.
Deswegen tue ich mich da schwer, das zu beschreiben, wie viele Leute da jetzt genau daran arbeiten.

[15:16] **Ok, kannst du trotzdem noch irgendeine Hausnummer geben?**

Ich weiß, dass die Entwicklungsabteilung, die für unsere Umfänge zuständig ist, da sind XXX Mitarbeiter, die direkt bei XXXXXXXXXXXXXXX arbeiten. Zusätzlich gibt es aber noch mal wahrscheinlich eine große Anzahl an externen Leuten, die damit arbeiten. Deswegen fällt es mir wirklich schwer, da jetzt eine genau Hausnummer zu sagen.

[16:39] **Wenn einzelne Designvariabel verändert werden, sind dann die Auswirkungen auf das Gesamtsystem transparent?**

Du meinst die Kausalitäten und Abhängigkeiten? **Genau**.

Ich glaube bei bestimmten Umfängen ist es bekannt, aber es gibt auch genug Fälle. Zum Beispiel, gerade bei dem Thema Software/Hardware, wo das getrennt voneinander entwickeln wird und die Abhängigkeiten nicht immer so deutlich sind.
Also ich glaube, für so traditionelle Umfänge, XXXXXXXXXXXXXXXXXX, wo es 10,20,30,40 Jahre lang Erfahrung gesammelt wurde, da ist, glaube ich, relativ gut beschrieben, wie sich Sachen verhalten. Aber bei neun Umfängen oder neun Teilen XXXXXXX, da sind die Abhängigkeit nicht immer sofort klar beschrieben und da kommt es auch schon vor, dass eine Entscheidung getroffen wird, die andere Sachen beeinflusst, aber man stellt das erst später fest.

[17:53] **Dann im späteren Entwicklungsverlauf?**

Ja, genau.

VARIANTENMANAGEMENT

[18:17] **Weißt du, wie viele XXXXXXXXXXXX es gibt und wie viele Varianten jeweils? (Mit denen du auch zu tun hast.)**

Das Ding ist, die Umfänge, die wir bearbeiten, sind eigentlich Baukästen. Das heißt, die sind über verschiedenste Produkte eigentlich gleich. Da bedeutet, da ist jetzt keine Varianz für die XXXXXXXXX, sondern das wird über die Produkte XXXXXXXXXX gleichmäßig eingesetzt. Es ist eher zeitlich, dass die alle paar Jahre aktualisiert werden. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Aber es gibt innerhalb dieses Baukastens bestimmte Varianten, zum Beispiel mit der XXXXXXXX, XXXXXXXXXX. Dann gibt es durch Länderspezifika induzierte Änderungen, das bedeutet, in China muss das System bisschen anders aussehen oder eine andere XXXXXXXXXXX zum Beispiel darauf laufen, als in Deutschland. Das sind schon mal Varianten, die entstehen.

[19:34] **Also Varianten entstehen eher durch die unterschiedliche Zusammensetzung aus den Baukästen und weniger, dass wir sagen, wir haben eine XXXXXXXXXX?**

Genau. Weil die Idee bei dem Baukasten, bei den Sachen, wo ich dran bin, ist, dass der über die verschiedenen Produkte gleich ist. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX dass man da den gleichen XXXXXXXXX und XXXXXXXXXXXXXXXXXXX hat und innerhalb dieses Baukastens der Kunde die Möglichkeit hat natürlich irgendwie Sonderausstattung zu wählen, die nochmal dafür Varianten bilden.

[20:12] **Also das heißt ja, dass in der frühen Phase schon Varianten berücksichtigt werden, die dort entstehen. Wenn ich es richtig verstanden habe, dann kannst du aus deinem Baukasten schon in der frühen Phase die Variante berücksichtigen?**

Ja, genau. Es wird in dieser Baukastenplanung natürlich frühzeitig geschaut, an welcher Stelle brauchen wir Varianz. Also bei der XXXXXXXXXX zum Beispiel brauchen wir unterschiedlich große XXXXX? Brauchen wir bei XXXXXXXXXXXXXX oder bei XXXXXXXXXXXXX unterschiedliche Größen? Und da, wo es relevant ist, werdenden frühzeitiger in der Entwicklung unterschiedliche Optionen erzeugt.
Auf Hardware Ebene aber auch auf Content Ebene, weil halt bestimmte Sachen durch regulatorische Vorgaben zum Beispiel, die wir hier in Europa machen, nicht in China gehen. Oder, weil zum Beispiel bestimmte XXXXXXXXXXX: Zum Beispiel das, was wir hier in Deutschland verwenden, kann nicht in China verwendet werden.

[21:24] **Wo wir wieder bei der Variantenbildung aufgrund von äußerer Komplexität sind.**

Ja. Ja, genau.

[21:31] **Wo sind da die Hauptprobleme bei dieser Auslegung mit dem Baukasten oder Variantenauslegung?**

Ich hab das jetzt noch nicht so oft durchlaufen…
Ich glaube, was schon eine zentrale Herausforderung ist im System zu definieren, was möglichst stabil, also, dass man die Teile stabil hält, die man in unterschiedlichen Ausstattungsgrößen verwenden kann in allen Ländern. Und nur dort Varianz zulässt, wo sie notwendig ist oder einen Mehrwert bringt. Ich glaube, diesen sozusagen Systemschnitt hinzukriegen. Ich glaube, das ist das Zentrale. Also wo brauche ich wirklich Varianz und wo soll ich sie sozusagen unterbinden? Ich glaube, das ist eine zentrale Herausforderung, gerade weil so viele Anforderungen von außen rankomme vom Kunden oder vom Gesetzgeber oder auch teilweise noch vom Markt selber, also von den Vertriebsorganisationen, wie sich bestimmte Sachen wünschen. Und da muss man natürlich schauen, was muss man berücksichtigen? Was ist nice-to-have? Und was erzwingen ich, dass da keine Varianz gibt.

[22:50] **Hast du da ein konkretes Beispiel, wo Varianz besteht, obwohl sie nicht notwendig ist?**

Hmm fällt mir jetzt spontan nichts ein.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Oder inwieweit, wenn ich bestimmte Sachen in dem einen Land anbiete, in dem anderen Land da mehr Hardware reinstecke, damit es keine Varianz gibt. Aber da sind wir noch in der Frühen Phase der Diskussion. Aber das sind zum Beispiel so Themen, wo ich mir vorstellen können, dass es da relevant wird.

ENTSCHEIDUNGSFINDUNG

[24:33] **Was fällt dir zum Thema Entscheidungsfindung bezüglich Herausforderungen ein in deinem Arbeitsleben oder aus Sicht eines Entwicklers?**

Ja genau, ich glaube die Entscheidungsfindung in der frühen Phase... Ich bin ja jetzt sehr viel in strategischen Entscheidungen und frühe Phase Entscheidungen mit eingebunden und in großen Organisationen, jedenfalls was mein bisheriger Eindruck ist, es gibt sehr viele Sichten auf dem Thema.
Es bedeutet da reden wahrscheinlich XXXXXXXXXXXX relativ oft über eine Entscheidung und jeder hat eine unterschiedliche Perspektive durch den Hintergrund. Den einen interessiert das Vertragliche, den anderen interessieren die Hardwareanforderungen, den anderen interessieren irgendwie die technische Ausgestaltungsmöglichkeiten. Das bedeutet, die zentrale Herausforderung, die ich sehe, ist alle reden über das gleiche Thema, aber jeder hat eine unterschiedliche Sicht drauf. Und wie kriege ich das hin, dass ich bei der Entscheidungsfindung die verschiedenen Sichten abgedeckt habe und dann auch für jede Sicht die Bewertung reinkriege, sodass ich das miteinander verweben kann und dann eine Entscheidung zu treffen. Ich glaube das ist so die zentrale Herausforderung. Weil jeder natürlich seine eigene Perspektive als besonders wichtig erachtet. Aber die Frage ist natürlich, wenn ich da 20 draufhabe, sind alle gleichwertig oder sind sie unterschiedlich wichtig? Und muss ich irgendwo Kompromisse eingehen?
Ich glaube, das ist eine zentrale Herausforderung, gerade wenn es darum geht, nicht evolutionär, sondern revolutionär Entscheidungen zu treffen. Also ich würde sagen, was zu machen, was es vorher noch nicht gab, ist natürlich schwieriger, weil man sich auch absichern möchte.

[26:28] **Und wie wird sich dann abgesichert?**

Was ich jetzt mitkriege, werden natürlich erst einmal viele Diskussionen geführt. Das bedeutet, es wird sich viel Zeit gelassen zu so einer Entscheidung. Was gut sein kann und was aber auch schlecht sein kann, weil es halt das langsam macht. Dann wird natürlich versucht, mit Bewertungs Frameworks und Entscheidungs Frameworks die verschiedenen Sichten rein zu bringen. Es werden auch XXXXXX zum Beispiel definiert, wo gesagt wird, wenn ich jetzt mit dem Partner über die genaue Ausgestaltung diskutiere, gehen wir bis zu dem Punkt in dem Thema, in dem Thema können wir Kompromisse machen, da haben wir eine rote Linie, die wir nicht überschreiten wollen. Man versucht, eine einheitliche Meinung zu definieren, bevor dieses Thema ausgestattet wird.

[27:29] **Okay, also ihr legt Entscheidungsrichtlinien fest?**

Genau, weil man natürlich immer Flexibilität hat, gerade wenn es jetzt wichtige Entscheidungen sind, die vielleicht auch über die nächsten Jahre die XXXX prägen werden und man dann mit einem starken Partner, der sozusagen auch eine starke Verhandlungsmacht hat in die Diskussion geht, dass man sagt: Ok, das wollen wir auf jeden Fall erreichen. Da sind wir zu Kompromissen bereit. Dass man auch ein scharfes Bild hat, was möchten wir eigentlich und was ist für uns wichtig?

[28:08] **Was fehlen für Informationen, um die Entscheidungen treffen zu können?**

Teilweise das technische Detailverständnis, welche Auswirkungen eine Entscheidung auf das Produkt hat. Ich merke das grade bei einer revolutionären Entscheidung, die gerade bei uns diskutiert wird. Das bedeutet, es wäre etwas XXXXXXXXX, was es vorher so nicht gab. Und dadurch, dass der Partner, mit dem man sich dann einigen würde, sehr stark ist, ist nicht klar, inwieweit das sozusagen XXXXXXXXXX einschränkt, inwieweit das Wettbewerbsvorteil oder Nachteile bringt, inwieweit man da in XXXXXXXX zum Beispiel läuft. Gleichzeitig ist aber auch nicht klar, wie sich das dann auf der technischen Seite manifestieren würde. Also inwieweit das Auswirkungen auf Entwicklungsumfänge XXX hat. Weil wir jetzt so ein Riesenpaket XXXXXXXX holen von einem externen und wir das vorher nicht extern bezogen habe, dann ist nicht klar, was wird dadurch beeinflusst? Inwieweit müssen wir unsere Arbeitsweisen zum Beispiel anpassen? Wie viele Freiräume haben wir noch?

[29:34] **Also es geht ja schon so ein bisschen in die Management Richtung und auch Ressourcenmanagement.**

Ja Ressourcenmanagement natürlich. Wenn ich etwas aus der Hand gebe und nicht mehr selber machen, sondern vom Partner beziehe, brauche ich natürlich weniger Leute, die das dann selber entwickeln. Aber die Frage ist, brauche ich denn mehr Leute, um das abzusichern, weil ich dann ja was von extern kriege, was ich nicht selber entwickelt habe, aber es muss ja trotzdem XXXXXXXXXX passen. Da habe ich dann schon wieder Ressourcen, die an anderer Stelle gebraucht werden.

[30:07] **Also auch die Entscheidung, dann geben wir es an einen externen Zulieferer oder nicht? Machen wir es lieber selber**

Genau. Ja auch ein Teil der Entscheidungsfindung.

ROBUSTES DESIGN

[30:48] **Werden die Produkte bei euch bezüglich robusten Designs bewertet?**

[Stille]

Oder was verstehst du denn unter robustem Design, damit wir da auf einem Level sind?

Also mein Verständnis von robustem Design wäre, dass ich bestimmte Änderungen vornehmen kann, ohne dass das dann Auswirkungen auf alle anderen Teile hat. Das wäre mein Verständnis. Das bedeutet, ich habe das teilweiße entkoppelt, teilweiße habe ich vielleicht aber auch Margen, dass sobald ich einen Parameter anfasse, nicht gleich das ganze System kaputtgeht oder ich das ganze System anpassen muss.

[31:32] **Hm, ja, das deckt sich auf jeden Fall mit meinem Verständnis.**

Sehr gut. (lacht)

[31:40] **Und wie sieht da jetzt die Bewertung aus von euren Produkten? Oder würdest du eure Produkte als robust bezeichnen?**

Dadurch, dass ich bisher nur in der frühen Phase war und nichts mit dem Thema Änderungen zu tun habe, fällt mir das ein bisschen schwer zu beurteilen. Aber ich weiß, dass wir auf jeden Fall viel Aufwand betreiben, um Anforderungen früh aufzunehmen, Abhängigkeiten zu erkennen, aber inwieweit das jetzt als robustes Design zählt, da fehlen mir die Einblicke ehrlich gesagt.

[32:18] **Würdest du aber Handlungsbedarf sehen ganz generell?**

Ich habe viel zum Thema Veränderungsmanagement XXXXXXXXXX davor gemacht. Also ich sehe da schon Bedarf, dass die Unternehmen da noch besser werden können. Weil es da natürlich irgendwie viele Abhängigkeiten gibt und auch Probleme, die noch nicht gelöst wurden.

Ich glaube, das geht auch so ein bisschen in Richtung Variantenmanagement, dass man als Unternehmen versteht, an welchen Stellen robustes Design auch benötigt wird. Ich glaube, das ist schon ein zentraler Punkt. Und da könnte man, so wäre meine Vermutung auch zum Beispiel noch auf Basis von historischen Daten noch besser lernen. Dass man sagt: Ok, ich werte aus, wie sind die Änderungen in der Vergangenheit aufgetreten? Was so Systeme zum Beispiel, die oft nochmal geändert werden müssen, wo wir zum Beispiel Flexibilität benötigen? Und ich glaube, da könne man auf jeden Fall noch weiter unterstützen und glaube, da können die Unternehmer auch noch besser werden.

ORIENTIERUNG

[33:37] **Möchtest du nochmal ein bisschen, was über das Thema Orientierung sagen, was du zum Anfang gesagt hast?**

Also ich denke eine zentrale Herausforderung, jedenfalls bei uns ist, die Entwicklung XXXXXXXXXXXXXXXXXX läuft relativ lange im Voraus an. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXX Aber, wie auch schon gesagt, wir haben jetzt viele auch in Richtung XXXXXXX Themenumfänge, wo sich eine hohe Dynamik sozusagen ergibt. Das bedeutet, wenn ich mir überlege, was ich in XXXXXXXXXXXXXXXXXXXX haben möchte, ob jetzt Dienste oder aber auch von Hardware, dann ist die Frage, wie stabil kann das überhaupt bleiben? Macht es Sinn, sowas so früh schon festzulegen? Weil halt einfach die Welt viel schnelllebiger ist als das, was ich mir jetzt vormachen, wenn ich sage, ich lege jetzt fest, was in fünf Jahren irgendwie XXXXXXXX läuft. Ich glaube, das ist schwierig.
Das bedeutet, in Richtung Orientierung kann ich halt noch schlecht sagen, was ich eigentlich XXXXXXXX brauche, um das Kundenbedürfnisse zu adressieren oder auch adäquaten mit einem Produkt darauf antworten zu können, was in die Zeit passt. Keiner weiß, 100 Prozent genau, was jetzt Richtung XXXXXXXXXX vielleicht XXXXXXXXXXXXXXXXXX, XXXXXXXXXXXXXX. Aber ich muss jetzt eigentlich schon Entscheidung treffen. Aber ich weiß nicht, wie sich gerade Innovationstrends zum Beispiel weiterentwickeln.

[35:20] **Ja, das klingt spannend. Ist jetzt aber wahrscheinlich aber eher ein Thema von Marktforschung, oder?**

Was ich meine natürlich. Aber Anforderungen, die ich definiere, hat ja auch damit zu tun, dass sich vorhersehe, was mein Produkt können muss. Wenn ich sage, da muss die und die Rechenleistung drin sein, zum Beispiel in XXXXXXXXXXXXXXXXXXXXXXXXXXXX, dann antizipiere ich ja sozusagen, was möchte ich denn da zeigen. Wenn ich jetzt aktuell nur XXXXXXX mache, dann ist es halt relativ easy. Aber wenn in Zukunft, XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX dann ist es halt nicht nur Marktforschung, sondern es hat halt auch mit der technologischen Entwicklung zu tun.

[36:05] **Die sich dann da anpassen müssen und eventuell flexibel bleiben sollten.**

Ja, genau. Das bedeutet, es ist halt schwierig, jetzt immer hundertprozentig zu sagen, was ich in der Zukunft möchte. Aber ich muss es eigentlich machen, um Entscheidungen zu treffen. Also es ist so dieser Spagat, den ich da sehe. Ich muss jetzt entscheidend, damit ich rechtzeitig fertig bin, aber ich entscheide, ohne genau zu wissen, wie die Zukunft aussieht, für die ich designe. Also da sehe ich so die zentrale Herausforderung. Ich muss irgendwas definieren, wo ich nicht hundertprozentig weiß, ob es wirklich so aussehen wird.

ANFORDERUNGSMANAGEMENT

[37:13] **Was fallen dir denn spontan für Herausforderungen in der frühen Phase ein, bezüglich den Anforderungen?**

Zentral denke ich einmal zu wissen, was sind Anforderungen, die ich in den verschiedenen Phasen definieren muss? Es gibt ja eine Hierarchie in den Anforderungen, das bedeutet, es gibt Anforderungen, die bereits viel von einem System festlegen. XXXXXXXXXXXXXXXXXXXXXXXXX Wenn ich da weiß, ich kann nur in diesem Teil mein XXXXXXX entwickeln, dann schreckt mich das schon mal zum Beispiel in vielen Sachen ein, weil ich nicht einen unendlich großen XXXX bauen kann. Wenn ich dann noch weiß, wieviel Gewicht mein XXXX hat oder was für ein XXXX dran ist, dann schränkt mich das noch weiter ein. Das bedeutet, es gibt so Toplevel Anforderungen, die man eigentlich kennen muss, die dann danach vieles schon vorbeschreiben. Das, denke ich, ist wichtig zu verstehen, was sind die Anforderungen, die das System Design treiben.
Dann natürlich die Abhängigkeiten zwischen den Anforderungen.
Und auch die Werte festzulegen für Anforderungen. Wenn ich weiß, ich muss zum Beispiel die XXXXXXX für XXXXXXXXXX bestimmen. Ist zwar gut, wenn ich weiß, das muss irgendwie XXXXXX sein, aber wie komme ich dazu? Wie weiß ich jetzt, was ich alles berücksichtigen muss, um den richtigen Wert zu finden?
Also noch einmal:
Was sind die zentralen Anforderungen, die ich gerade zu Beginn entscheiden muss, damit ich dann halt mich langsam runter entwickelt kann bei den Anforderungen. Wie hängen die Anforderungen zusammen? Und dann wie fülle ich die Anforderungen mit Werten? Das sind die zentralen drei Herausforderungen.

[39:23] **Eine Einteilung auf verschiedene Anforderungsebenen, die wird auf jeden Fall schon berücksichtigt und gelebt im Unternehmen? Also, dass man sagt, man hat verschiedene Produktstrukturen, Komponenten-, Systemebenen.**

Genau, es gibt Basisanforderungen, dann gibt es Teileanforderungen und so was. Es wird dann natürlich vom großen Ganzen erstmal, was sozusagen die Grundlagen schafft, angefangen. Und dann wird das immer weiter runter gebrochen.

[40:07] **Wie viele Anforderungen sind ungefähr quantitativ erfasst, prozentuell?**

Ich habe bis jetzt noch keine Anforderungsliste ausgefüllt. Aber ich vermute mal, es wird sehr viel sein. Das hängt wieder davon ab, was ich mir anschaue. Wenn ich XXXXXXXX angucke, da gibt es ganze Abteilungen, die sich nur mit dem Thema Anforderungsmanagement beschäftigen.

Für die Umfänge, mit denen ich zu tun hab, habe ich noch keine Anforderungslisten befüllt oder bewertet.

[40:55] **Dann noch die Frage, wie viele Anforderungen in dem Produkt durchschnittlich gestellt sind. Jetzt sagen wir mal, auf der obersten Ebene. Also die Basisanforderungen. Hast du da eine Ahnung?**

Also ich glaube, wenn ich da irgendeine Zahl sagen würde, dann wäre das einfach geraten.

[41:36] **Wie häufig treten Zielkonflikte in der frühen Phase auf? Wenige Male, hin und wieder oder bei fast allen Anforderungen?**

Hin- und wieder. Es gibt manche Sachen, wo ich glaube, dass relativ schnell eine Entscheidung getroffen werden und sich alle im Klaren darüber sind. Aber dann gibt es andere Anforderungen, wo viel diskutiert wird, weil wieder die Sichten unterschiedliche sind und auch die Anforderungen von den Abteilungen verschieden sind.

TEIL 2

[53:12] **Wo würdest du einen Anwendungsfall für die Forschungsansätze im Unternehmen sehen?**

Also die Frage wäre jetzt wo kann ich diese Untersuchungen einsetzen, um festzustellen, wie ich mein System am besten konstruieren muss, damit es kostenoptimiert ist, oder?

[53:59] **Kannst du das nochmal wiederholen?**

Also du möchtest jetzt ein Beispiel dafür haben, für diese Untersuchung mit den Kosten über Komponenten, wo man das bei uns einsetzen könnte?

Genau. Wo dir einfällt bei euch im Unternehmen, wo das fehlt, wo man das an einsetzen könnte.

Ich vermute mal, dass man es in Bereichen, wo viele Varianten vorhanden sind. Vielleicht im Bereich XXXX, dass man das ist auf jeden Fall da einsetzen könnte.
Ich meine, es macht ja nur Sinn es irgendwo einzusetzen, wo ich viel Varianz habe und die eigentlich nicht haben möchte, oder?

Ja. Wo man das Gefühl hat, dass Kosten unnötig über Varianten entstehen.

Da wüsste ich zum Beispiel XXXXXXXXXXX, also die ganze Vernetzung XXXXXXXXXXX. XXXXXXXXX, die die einzelnen Bauteile miteinander vernetzen. Ich glaube, da könnte man es einsetzten, weil das im Moment sehr XXXXXXXXXXX individuell ist und das viel Aufwand erzeugt.
XXXXXXXXXXX...

[55:30] **Werden denn generell so Standardisierungen betrachtet in der Entwicklung also nach dem Grad der Standardisierung gefragt?**

Ich vermute mal schon. Aber ehrlich gesagt habe ich da jetzt keine detaillierten Einblicke, wie das genau abläuft. Aber ich weiß zum Beispiel, dass wenn man sich XXXXXXXXX neu konfiguriert, nicht alles frei wählen kann. Das bedeutet, es werden bestimmte Kombinationen schon vorgegeben. XXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX, damit man sozusagen nicht alles frei wählen kann, sondern so bestimmte Packages vordefiniert werden.

[56:07] **Wo siehst du denn die Stärken von dem Ansatz der kostenoptimierten Produktfamilien Auslegung bei euch im Unternehmen?**

Ich muss ehrlich sagen, da fehlen mir ein bisschen der Einblick jetzt, wie das genau gemacht wird. Und ob die schon solche Ansätze verwenden oder ob weiterer Handlungsbedarf bestehen. Da bin ich schon zu weit weg davon.

[56:54] **Aber es gibt Potential den Ansatz [Produktfamilienauslegung] anzuwenden?**

Ja. Ja, vorstellen, kann ich es mir. Ich kann halt schlecht bewerten, was die da jetzt gerade schon machen.

[57:10] **Wie sieht es denn mit Kostenmodellen aus? Also gerade was Herstellkosten, entstehende Komplexitätskosten... Gibt es da Modelle?**

Da gibt es eine Fachabteilung, die sich nur mit Kostenschätzung beschäftigt und sozusagen dann auch ein Richtwert ausgibt, wo man sagt, ob das wäre jetzt zum Beispiel ein Preis, zu dem man dieses Produkt oder diese Komponente kaufen sollte. Das bedeutet es gibt da Kostenmodelle. Ich glaube, diese Modelle sind parametrisch, das bedeutet es gibt bestimmte Eingangsgrößen, wo gesagt wird die beeinflussen die Kosten zum Beispiel. Es wird der Herstellungsprozess mitberücksichtigt. Logistik wird auch berücksichtigt. Also da gibt es was, ich weiß wieder nicht im Detail, wie ausgefuchst und wie kompliziert das alles ist.

[58:11] **Mhm ok, aber es ist ausgelagert?**

Es gibt eine zentrale Fachabteilung, die sich nur mit dem Thema Kostenschätzungen beschäftigt.

[58:26] **Gehen wir nochmal zurück zur dem anderen Forschungsansatz, zu den Datenbanken. Siehst du da vielleicht auch einen Anwendungsfall? Es ist ja speziell auch bei XXXXXXXXXXX interessant, ist jetzt natürlich nicht so genau dein Gebiet, aber eben auch so etwas wie XXXXXXXXXXXXX.**

Ja, Modellerstellung denke ich macht Sinn. Ich könnte es mir auch vorstellen, ist jetzt ein anderes Thema. Aber ich glaube, dass das bei uns im Bereich Software gemacht wird. Wobei man da ja Abhängigkeiten von Funktionen hat, das ist ein bisschen was anderes. Aber ich mein grundsätzlich ist es interessant und relevant. Ich kann halt nicht einschätzen, was bei der Entwicklung und der Auslegung der Komponenten gemacht wird.

[59:18] **Es ist jetzt für dich speziell in deinem täglichen Arbeitsumfeld auch kein Thema von Arbeitserleichterung oder sowas?**

Ich bin in strategische Entscheidungsprozesse eingebunden, da sind wir ehrlich gesagt noch gar nicht so tief auf den technischen Anforderungen. Dass wir da jetzt über sowas diskutieren, wo wir Zielgrößen und Stellhebel im Detail besprechen. Es geht eher so ins global Galaktische worüber wir reden und entscheiden.

1:00:04 **Gut, danke für deine Einblicke.**

Nr. Interview	02	Unternehmen	X	Datum und Ort	26.05.2020, München	Interviewer	XXX
Skype Meeting, audio- und visuell, keine Störungen und angenehme Atmosphäre. Dauer: 01:27:28							

[02:09] **Würdest du dich bitte kurz vorstellen und deine Tätigkeit im Unternehmen beschreiben?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[02:47] **Und seit wann bist du auf dieser Position tätig?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[03:21] **Also deine Tätigkeit an sich hat sich nicht verändert nur das Einsatzgebiet?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

BEGINN HAUPTTEIL

[04:07] **Fallen dir spontan noch weitere (Herausforderungen) ein, wenn du jetzt an Herausforderungen in deinem Arbeitsleben denkst?**

Es gibt natürlich vielfältige Herausforderungen. Ich versuche, das jetzt gerade noch einzuordnen.
Anforderungsmanagement ja, definitiv ist eine Herausforderung.
Entscheidungsfindung: Müsste man vielleicht näher klären, was damit genau gemeint ist. Würde ich jetzt erstmal nicht rauspicken.
Variantenmanagement ebenfalls eigentlich nicht.
Komplexität: Müsste man vielleicht auch noch mal näher darauf zu sprechen kommen. Aber würde ich grundsätzlich zustimmen.
Robustes Design als Zielsetzung jedenfalls ebenfalls.
Weitere Herausforderungen würde ich vor allem noch da drin sehen: Ein Aspekt sehe ich vor allem in Kompatibilität. Also wir haben bei vielen Produktentwicklung schon immer mit Kompatibilitätsaspekten zu tun, die die Lösungsräume doch deutlich einschränken können und das ist nicht ganz ohne.
Eine große Herausforderung, die wir noch haben ist natürlich die Benutzerfreundlichkeit. Also, das betrifft einfach viele von unseren Produkten. Diese scheitern manchmal eben auch, wenn sie nicht benutzerfreundlich sind und nicht intuitiv sind.

[05:43] **Das ist ja aber eigentlich ein Teil der Anforderungen. Und wird wahrscheinlich als Anforderung formuliert?**

Ja, würde jetzt mal so grob als Anordnung sicherlich formuliert werden. Aber dann stellt sich ja immer noch die Frage, wann wird die Anforderungen erfüllt? Wie ist es also zu formulieren und wird da abgewägt? Ist es eine Ja/Nein Entscheidung? Das wäre alles ein Aspekt.
Eine weitere Herausforderung sehe ich noch in Kooperationen. Also gerade, wenn wir von XXXXX sprechen haben wir es immer auch mit Partnerfirmen zu tun. Und das ist auf jeden Fall auch eine Herausforderung. Wobei man fragen muss, sprechen wir von Projekt Herausforderung oder technischer Herausforderungen?

[06:31] **Also Kooperationen können wir auf jeden Fall mit aufnehmen. Das ist nämlich auch was, was gut in den Ansatz passt.**

Jetzt muss ich noch einmal gerade überlegen. Ich habe mir da auch noch einmal Notizen gemacht zu den Herausforderungen allgemein.

Anwendungszuverlässigkeit ist auch eine große Herausforderung. Die Funktion unserer Produkte hängt schon immer zu großen Teil vom Anwender ab. Dreht er das fest genug zu oder eben nicht, dreht er das zu fest zu. Den schwächsten Anwender irgendwie zu berücksichtigen, ist für uns schon immer eine Herausforderung.

[07:21] **Da springt mir das mit dem Robusten Design gerade ins Auge.**

Richtig, das passt. Ansonsten habe ich die Herausforderung Alterung. Das ist immer wieder ein Thema aus verschiedenen Gründen. Und letztendlich hätte ich nur noch das Thema Technologie als Herausforderung, wo wir auch merken, dass das ein Aspekt ist.

[07:49] **Kannst du den letzten Punkt mit der Technologie noch mal kurz ansprechen. Was hast du damit genau gemeint?**

Ich habe bestimmten Projekten bestimmte Herausforderungen zugeordnet und da ist aufgefallen, dass manche Produkte eben auch als Herausforderungen hatten, dass die Technologie nicht so ganz etabliert war. Es kann sowas einfaches sein, wie ein Faserverbund lackieren zu wollen. XXXXXXXXXXXXXXXXXXXXX. Da war halt die Technologie, für uns jedenfalls noch nicht, und für unsere Zulieferer auch noch nicht vorhanden.
Andere Aspekte gibt's vielleicht auch bei anderen Produkten. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXX. Das war auch eine Technologie Herausforderung. Erstmal, weil es in der Zuliefererkette und auch bei uns noch nicht komplett beherrscht war. Und sowas kommt doch häufiger vor, weil wir uns halt wirklich nicht einschränken lassen wollen und deswegen oft das auch als Herausforderung direkt im Projekt haben.

[09:01] **Und die Herausforderung ist dann hauptsächlich dem geschuldet, dass es die Technologie so für den Anwendungsfall noch nicht gibt? Oder ist es eher, dass man nicht sicher weiß, welche Technologie verwendet man am besten oder sich da zwischen verschiedenen Verfahren entscheiden muss?**

Düfte beides sein. Zum einen haben wir sicherlich am meisten damit zu kämpfen, dass wir dann in einer Situation sogar schon sind, dass ein bestimmter Faserverbund lackiert werden soll. Und damit nehmen wir uns die Möglichkeiten, den Faserverbund fürs Lackieren zu optimieren, weil der schon sozusagen entwickelt ist. Da kann man natürlich im Vorfeld schon schauen, dass die Alternativen besser bekannt sind. Aber auch danach ist es einfach, so, dass es, wie gesagt, zumindest für die uns zur Verfügung stehenden Lieferketten auch fehlt, dass da die Erfahrung da ist oder eben auch die Technologie wirklich beherrscht wird mit den Werkzeugen und Maschinen. Die gibt es aber bestimmt für andere Firmen, also große Firmen. Wie XXXXX, die haben solche Technologien wahrscheinlich schon seit zehn Jahren dann im Zweifelsfall, aber die ist dann halt auch nicht als Service für andere verfügbar.

[10:24] **Ja, also der Zugang dann auch zu anderen Technologien.**

Genau.

BEGINN HAUPTTEIL

[11:44] **Kannst du die (sechs Themengebiete) in eine Reihenfolge bringen? Es geht darum, dass wir auf diese sechs Themen jetzt nochmal genauer eingehen und so die Priorisierung gesetzt wird.**

Magst du nochmal kurz was zur Komplexität sagen?

Also, da geht es viel um die Produktkomplexität. Also wie hängen verschiedene Designvariablen miteinander zusammen, wie hängen Komponenten miteinander zusammen. Wie ist die Struktur des Produkts

Okay, ich glaube das ist bei uns eher nicht so ausgeprägt. Oder das ist nicht die Regel, dass das eine der wesentlichen Herausforderungen ist.

Ich mach das jetzt auch ein bisschen aus dem Bauchgefühl.

1. Anforderungsmanagement 2. Kooperationen 3. Robustes Design 4. Entscheidungsfindung 5. Komplexität 6. Variantenmanagement

---

ANFORDERUNGSMANAGEMENT

[13:59] **Also es wird jetzt hauptsächlich immer darum gehen, welche Herausforderungen entstehen. Was sind die Ursachen? Und wie wird aktuell damit umgegangen bei euch im Unternehmen.**

[14:18] **Also wir reden ja immer noch über Herausforderungen in der frühen Phase und gerade da ist die Erstellung von Anforderungen ein zentrales Thema. Welche Herausforderungen treten bei euch im Unternehmen in der frühen Phase denn auf?**

Also eine Herausforderung ist sicher schon mal überhaupt eine vernünftige Basis für Anforderungen zu haben. Also erst einmal das Zusammenstellen der Anforderung. Das muss man sich in der Regel erarbeiten, das wird einem nicht präsentiert. Und die zweite Herausforderung dabei sehe ich auch darin, die beteiligten Personen dazu zu motivieren, Zeit zu investieren.

[15:22] **Also die beteiligen Personen sind die Entwickler?**

Ne, gar nicht unbedingt, sondern eher das, was wir dann eben Steering Commitee oder Stakeholder nennen oder auch Produkt Manager, die oft im Steering Commitee auch mit drinsitzen, dass die wirklich da am Ball bleiben und nicht mal eben die Anforderungen übern Zaun werfen und sagen das war's jetzt. Weil da ist es meistens ja sehr, sehr oberflächlich noch alles. Und deswegen ist es wichtig, dass damit Zeit eingeräumt wird und auch die Bereitschaft da ist und erkannt wird, dass das einen Mehrwert bringt da nicht nur einmal teilzunehmen, sondern immer wieder teilzunehmen.

[16:05] **Werden die Anforderungen vollständig definiert in der frühen Phase oder wird da noch viel verändert im Laufe der Produktentwicklung?**

Ja die Anforderungen werden schon vollständig definiert, wobei auch Änderungen der Anforderungen durchaus willkommen sind. Ich sehe jetzt kein Grund hier Änderungen der Anforderungen abzulehnen. Man muss halt die Konsequenzen sehen, die sich dann ergeben. Obwohl Anforderungen abgeschlossen werden oder der Katalog schon bestimmt ist, laufen diese Anforderung Gefahr, dass sie sich noch ändern können. Sind die Final oder können die gar nicht ausgefüllt werden? Sprich, hat man da noch gar nicht die notwendige Entscheidungsgrundlage? Da muss natürlich dann gezielt darauf hingearbeitet, dass man da die Antworten darauf findet.

[17:27] **Du hast vorher noch mal kurz das Thema Stakeholder angesprochen. Das ist eine Quelle von Anforderungen. Wo kommen sonst die Anforderungen her?**

Du spielst darauf an, ob die vom Kunden, oder? Also tatsächlich Ist das nicht direkt vom Kunden. Wir sprechen jetzt ja über den Regelfall. Es ist nicht direkt, dass die Anforderungen bei uns vom Kunden kommen, sondern die werden indirekt weitergeleitet, zum Beispiel von den Produktmanagern. Was die eben für Feedback bekommen. Auch offizielles Feedback das bei uns in XXXX hinterlegt ist. Also tatsächlich ist da ein bisschen Stille Post natürlich schon beteiligt in dem Fall.

[18:29] **Wie werden die Anforderungen gesammelt. Was gibt es für Tools und Werkzeuge bei euch?**

Grundsätzlich haben wir ein eigenes Tool, XXXXXXXXXXXXXXXXXXXXXXXX.
Ja, also es geht eher ein bisschen darum, ob ihr sowas wie Produkt- und Funktionsstrukturen und sowas schon abbildet in den Anforderungen oder ob das Listen, in denen einfach gesammelte Herausforderungen unsortiert stehen oder sind die bereits zum Beispiel mit Produktfunktionen verknüpft?
Nein, mit Produktfunktionen wird's eigentlich jetzt per se nicht verknüpft. Nein.

[19:30] **Und wie sieht es mit der Formulierung der Anforderungen quantitativ aus? Wie viele Anforderungen werden quantitativ erfasst. Kannst du dazu eine Prozentangabe geben?**

Grundsätzlich alle. Also das Ziel jeder Aufforderung ist, diese quantitativ zu erfassen. Wobei wir ja schon bewusst da einen Zwischenschritt gehen, um eben auch bewusst die Anforderungen nicht unnötig ein zu bremsen.

Wo wir konkret werden, was mir sofort einfällt, ist das Produkt muss kompatibel XXXXXXXXXXXXX XXXXXXXXXX sein. Und da wird natürlich meistens immer lange rumgedruckst. Ja, welches soll das denn sein? Von welchem Hersteller? Welches Modell? Und um das abzukürzen, versuchen wir immer, diese Anforderungen erstmal so zu definieren, dass es mit einem sehr weit verbreiteten XXXXXXX kompatibel ist. Damit ist nämlich der Produktmanager eigentlich schon mal zufrieden und dann weiß man schon mal, jetzt braucht man eigentlich nur noch herausfinden, was ist denn ein weit verbreitetes, XXXXXXXXXX. Und das wird dann später natürlich auch in die Anforderungen reingenommen und überführt.
Das Ziel ist immer da konkret zu werden. Auf jeden Fall.

[21:35] **Wie häufig treten Zielkonflikte in der frühen Phase? Ich habe drei Möglichkeiten zur Auswahl: wenige Male hin, hin und wieder oder bei fast allen Anforderungen.**

Ich würde sagen, eher nicht so häufig.

Also hin und wieder oder wenige Male?
Was ist jetzt davon das Häufigere? (lacht) hin und wieder ist häufiger? **Ja.** Gut, nehmen wir hin und wieder.

Also genau, der Konflikt zwischen Anforderungen ist vielleicht gar nicht so ausgeprägt wie erwähnt. Was allerdings schon mal vorkommt, ist, dass wir einfach Anforderungen aus verschiedenen Quellen haben, zum Beispiel aus dem US-amerikanischen Markt und dem europäischen Markt. Dann vielleicht auch noch aus unterschiedlichen Anwendungen bei uns heraus. Und die formulieren ja nicht alle die gleichen Anforderungen, und das ist in gewisser Weise auch ein Konflikt. Nicht notwendigerweise jetzt zwischen verschiedenen Anforderungen, aber doch, der eine braucht ja vielleicht etwas anderes als der andere. Und das droht immer ein bisschen verloren zu gehen, weil erst mal alles in einen Topf geschmissen wird und einfach nicht getrennt nach Märkten die Anforderungen gesetzt werden und dann geschaut wird, was ist die Schnittmenge. Gibt es eine Schnittmenge? Wo gibt es sie nicht?
Also das finde ich schon einen wichtigen Punkt. Vor allem, weil man sonst oft in der Situation kommt, wo man mit Anforderungen eben zu keinem vernünftigen Ergebnis kommt, weil dann jeder den Kompromiss eingeht und man gar nicht weiß, wie groß der Kompromiss wirklich ist.

[23:59] **Das bezieht sich aber wahrscheinlich hauptsächlich auf Testverfahren und sowas oder bezieht es sich auch wirklich auf die Auslegung der Produkte. Also, dass dann Varianten entstehen durch unterschiedliche Regulatorien in den verschiedenen Ländern?**

Nein, es betrifft eigentlich gar nicht die Märkte oder die XXXXXXXXXX oder dergleichen, sondern tatsächlich auf die Frage wie soll das Produkt ausschauen? Also sagen wir es so der Amerikaner wünscht sich halt ein Pickup und der deutsche vielleicht ein Porsche. Und wenn jetzt alle Anforderungen in einen Topf kommen, dann entsteht etwas, was eigentlich keiner mehr fahren will. (Und bei uns ist halt immer das Ziel, eigentlich keine Variante.)

VARIANTENMANAGEMENT

[25:03] **Du meintest ja gerade, dass ihr versucht Varianten zu vermeiden. Aus welchem Grund?**

Das ist jetzt allerdings eine gute Frage. Ich glaube, da sind wir natürlich so geprägt, dass wir relativ viele Varianten schon haben bei verschiedenen Produkten älterer Art. Wir haben einfach auch historisch schon verschiedene Varianten. Man hat ein neues Produkt entwickelt, das Alte gilt dann als Variante und gegenüber den Kunden ist es halt mutmaßlich nicht immer die beste Lösung und auch im Verkauf wahrscheinlich nicht förderlich. Ich kann's nicht ganz benennen.
Aber ich glaube, die wesentliche Begründung ist einerseits Gewohnheit. Das ist ja der Ausgangspunkt und jetzt muss man an der Stelle schauen, wie begründet sich das jetzt gut oder nicht. Und da würde ich sagen, es wird dann der Aufwand als geringer eingeschätzt. Der Entwicklungsaufwand und auch das Portfolio, was wir anbieten, ist für den Kunden stimmiger, weil einfacher. Er muss dann nicht irgendwie groß aussuchen.

Abschließend muss man vielleicht noch sagen unsere Stückzahlen sind generell gering und mit Varianten wird das natürlich drohen nochmal komplizierter zu werden. Also teurer im Einkauf und letztendlich entsteht mit jeder Variante auch die Notwendigkeit, bestimmte Dinge wie zum Beispiel Handbücher und vor allem sowas wie einen Platz XXX XXXXXXX vorzusehen. Also das, was dann on Top kommt, ist das, was so ein bisschen abschreckend wirkt.

Aber insgesamt stimmt schon, man muss vielleicht auch mal überlegen, ob es nicht eine Varianten-Möglichkeit geben darf.

KOOPERATIONEN

[27:28] **Kooperationen hattest du als zweites genannt, magst du da nur mal kurz was dazu sagen?**

Die Herausforderung bei Kooperationen ist natürlich immer, dass man mit anderen Parteien zusammenarbeitet. Die sich erst mal in ihrem eigenen Kosmos befinden, sprich ihre eigenen Gewohnheiten haben und teilweise überzeugt werden müssen, um Dinge zu erreichen. Das ist natürlich eine ähnliche Überzeugungsarbeit wie intern, aber von einem ganz anderen Standpunkt häufig ausgehend.
Und dann kommt zusätzlich noch die Schwierigkeit der zeitlichen Planung dazu. Auch zum Teil, dass man eben nicht genau weiß, was die für Vorstellungen haben, welche Ziele die wirklich konkret verfolgen. Und auch entsprechend nicht in ähnlicher Tiefe Einblick hat über all diese Themen.
Also das Verständnis über Firmen hinweg, das gemeinsame Produkt oder das jeweilige zugehörige Produkt zu verstehen und dort auch einen ähnlich getakteten Entwicklungsprozess zur haben, in den man gut Einblick hat, das ist schon eine große Herausforderung.

[28:58] **Wie viele dieser Kooperationsschnittstellen gibt es durchschnittlich bei einem Produkt?**

Beteiligte Firmen grundsätzlich eins, maximal zwei, würde ich sagen.

ROBUSTES DESIGN

[29:35] **Was verstehst du denn unter robustem Design?**

Robustes Design, da fällt mir zunächst eher ein, dass das ein technisch äußerst zuverlässiges Produkt ist. Es geht nicht kaputt. Und es kennt keine Zustände, die zu Unsicherheit XXXXXXXXXXXXX beim Anwender führen. Wobei, das ist genau der Bereich, wo ich bei robustem Design sofort auch unterscheiden wollen würde zwischen Bedienerfreundlichkeit oder Useability. Das ist für mich schon ein anderer Aspekt. Du hast es jetzt eigentlich zusammengeführt.
Genau. Wir sehen das ein bisschen allgemeiner. Im Prinzip ist die Definition bei uns, dass ein robust entwickeltes Produkt selbst bei Varianz oder bei starken Abweichungen vom Sollwert oder bei Toleranzen nicht seine Anforderungen verletzt. Aber Anforderungen sind ja auch eigentlich so etwas wie Benutzerfreundlichkeit und so.
Tatsächlich ist das auf jeden Fall ein Thema. Die Herausforderung bei uns ist ja, ein besonders XXXXXXXXXXXXX, wo findet man den? Also was für ein Drehmoment bringt er auf und genügt es dann noch? Das ist ja eigentlich dann dieses Beispiel.

[31:16] **Und werden die Produkte frühzeitig danach bewertet, ob es ein Robustes Design ermöglicht?**

Ja, das schon.
[31:32] **Und siehst du da noch Lücken oder Handlungsbedarf?**

Ja, ich würde trotzdem Lücken sehen, weil ich ja sehe, dass da Schwierigkeiten sind und die sich derzeit auch nicht immer restlos auflösen lassen und auch mal Fehleinschätzungen denkbar sind.

[31:48] **Und das führt dann zu welchen Problemen im späteren Prozess? Kannst du da ein Beispiel nennen?**

Wenn es eine Fehleinschätzung gab, dann kannst du natürlich erforderlich sein, dass das Produkt geändert werden muss und umgestaltet wird.

[32:02] **Und das ist mal mehr, mal weniger Aufwand?**

Absolut. Dann kommen gerade wieder diese Kompatibilitätsthemen wieder auf. Dann hat man ja meistens wieder den Kosmos geschaffen oder sogar erweitert, mit dem man kompatibel sein muss. Und muss dann aber auch noch ein robustes Design möglichen.

ENTSCHEIDUNGSFINDUNG

[33:09] **Bei der Entscheidungsfindung geht es hauptsächlich darum, dass während der Konzeptphase viele Entscheidungen getroffen werden müssen, gerade bezüglich des Designs. Wir würden gerne wissen, wie werden diese Designentscheidungen getroffen und was für Herausforderungen entstehen da?**

Wenn es mehrere Konzepte gibt, was anzustreben ist, dann wird natürlich eine Bewertung vorgenommen. Da gibt es verschiedene Ansätze, und keins davon ist extrem überzeugend.

Zum einen kann man natürlich mit einem morphologischen Kasten arbeiten, um da schon mal eine Vorbewertung zu machen. Wenn es aber konkrete Lösungen gibt, dann nutzen wir eigentlich recht gern das House of Quality, um das gegenüber den Anforderungen, mit einer bestimmten Gewichtung, dann in eine schöne Zahl zu überführen. Das hat natürlich alles Grenzen. Es hängt dann auch wieder davon ab, wie das House of Quality aufgebaut ist. Dann kommen natürlich auch wieder andere Themen raus.
Was man dann natürlich auch macht, ist, dass man verschiedene Lösungen einfach paarweise gegeneinander vergleicht. Welche ist jetzt besser? Und aus diesen paarweisen Vergleichen leitet sich danach auch wieder ein Ergebnis ab. Also, da gibt es verschiedene Werkzeuge. Es gibt jetzt keine klare Vorschrift. Es muss House of Quality sein. Es muss mit dem paarweisen Vergleich gemacht werden. Oder ist überhaupt der Lösungsraum ausreichend, Stichwort: morphologischer Kasten oder solche Themen. Da gibt es keinen klaren Weg, der genommen wird. Es hat

letztendlich schon viel damit zu tun, dass Personen beteiligt sind, die über Erfahrungen verfügen und dann eben die Entscheidung treffen.

Entscheidungsfindung. Warum habe ich das vielleicht gar nicht so beleuchtet? Weil ich behaupte, es klappt ganz gut. Und die Findung ist ja auch immer die Frage können wir uns entscheiden, trauen wir uns zu entscheiden? In die Richtung habe ich tatsächlich eher gedacht. Und da tun wir uns nicht schwer. Wir drucksen eigentlich jetzt nicht wahnsinnig rum. Natürlich gibt es Situationen, wo man sagt, wir sind noch nicht an dem Punkt eine Entscheidung treffen zu können. Wir brauchen mehr Varianten, wir müssen noch das Thema beleuchten. Das gibt es schon. Aber das hat ja dann auch eine klare Zeitschiene, die dann noch fehlt.

[36:28] **Da ist noch eine Frage, und zwar der Anteil an Simulationen und Berechnungen bei der Entscheidungsfindung. Wie hoch ist der denn in etwa? Also werden schon frühzeitig Simulationen und Berechnungen zur Entscheidungsfindung hinzugezogen oder eher später?**

Generell würde ich sagen, dass der Anteil von Simulationen und Berechnungen eher gering ist, unabhängig vom Zeitpunkt.
Wobei gerade der Aspekt wichtig ist erstmal zu verstehen, wie die Physik dahinter funktioniert. Was für Kennzahlen gibt's überhaupt und dass dann zumindest überschlagsweise oder per Experiment einzuordnen. Also, da kann es ja darum gehen, muss der Flächenkontakt groß sein oder genügt es ihn klein zulassen? Braucht man da eine kleine Flächenpressung oder eine große? Da geht es um die Gleichmäßigkeit der Flächenpressung.
Das ist ein Reibungsproblem, das ich da konkret aufgreife. Das ist natürlich schon wichtig, dass man sich bewusst macht welche Kennzahlen da sind und ob jetzt eine Kennzahl eher nach oben oder unten getrieben werden muss. Ob sie jetzt bedeutend ist oder nicht so bedeutend ist, das findet schon statt. Aber tatsächlich muss man das schon puschen. Und vor allem auch, dass die Physik verstanden wird, muss man pushen. Dann bin ich aber schon echt zufrieden, wenn ich sehe, dass sich da Gedanken gemacht wurde und auf Papier grobe Überschlagsrechnung oder einfachste Versuche gemacht worden sind.
Simulation jetzt wirklich, dass der Computer angeworfen wird: Eher nicht, weil das bei uns eigentlich bedeutet, dass wir sowas wie FEM machen, um die mechanische Stabilität zu beleuchten. Und die ist eigentlich in der frühen Phase selten irgendwie von Bedeutung.

…

TEIL 2

[53:51] **Hast du Fragen zu den gerade vorgestellten Forschungsansätzen. Beziehungsweiße hast du verstanden, was ich dir erklärt habe?**

Absolut. Die große Frage war ja wo kommen die Zusammenhänge her? Wie zeitintensiv ist das Ganze? Aber das soll ja mit diesen vorgefertigten Modulen abgefangen werden.
Genau.
Oder, wenn man natürlich ähnliche Fälle hat, dann reduziert es die Zeit de facto erheblich.

Also interessant ist es auf jeden Fall. So manche Sachen, wie kostenoptimierte Familienauslegung, sehe ich jetzt bei uns noch nicht. Das hat damit zu tun, dass wir in der Regel komplette Produkte bekommen und nicht die einzelnen Komponenten dann montieren, sodass dieser Sachverhalt immer nur gedämpft bei uns ankommt, wenn wir da Optimierungen machen würden. Aber das kann sich in Zukunft auch ändern und das ist auch nur ein Aspekt des Ganzen.
Insgesamt deckt es sich schon gar nicht mal so schlecht mit den Aspekten, die wir schon machen. Aber unser Variante ist die langsame Papiervariante. Aber die finde ich eben auch wichtig, weil man muss sich ja mit den Themen auseinandersetzen und die Abhängigkeiten verstehen. Aber die Leistungsfähigkeit ist hier natürlich schon deutlich größer.

[55:31] **Wo siehst du denn besonders die Stärken bei der Anwendung beim Unternehmen XX? Fangen wir mal mit der automatisierten Modelldarstellung an.**

Ich kann mir noch nicht ganz vorstellen, was da jetzt alles dann möglich sein wird. Das hängt natürlich auch von den Modulen ab.

Du kannst sowohl Stärken als auch Schwächen nennen.

Die Schwäche ist natürlich immer: Ich muss ja erst mal wirklich die Daten und die Zusammenhänge haben. Wenn mir die präsentiert werden und wenn ich sie dann richtig verknüpfen kann, dann ist es natürlich ein sehr starkes Tool. Aber genau davon hängt es ab. Was sind meine quantifizierten Zusammenhänge und sind die Zusammenhänge

überhaupt richtig gewählt und vergesse ich dabei auch nichts Entscheidendes? Sonst beeinflusse ich wahrscheinlich erheblich mein Ergebnis.
Ich glaube nicht, dass es jetzt ein Thema ist, wo man sagt jetzt ersetzen wir vielleicht eine Vorgehensweise, die wir schon haben, in einem Fall, wo wir uns einigermaßen komfortabel oder gut aufgehoben fühlen. Es muss also eher schon was sein, wo wirklich die Not größer ist, um jetzt auch diesen Mehraufwand anzugehen.

[57:03] **Also du siehst, dass das gerade ein größerer Mehraufwand wäre, als es dann tatsächlich Nutzen bringt?**

Nein, nicht unbedingt. Aber ich glaube, das wird dann der tatsächliche Eintrittspunkt sein zu sagen, wann würden wir das machen? Vor allem in den Bereichen, wo wir eben mit dem klassischen Vorgehen Schwierigkeiten erleben. Es kann ja diesen Entwicklungsschritt durchaus beschleunigen, wenn es eben ein wie vorher gezeigt, so ein iteratives annähern an die richtige Lösung ist. Aber genau darauf kommt es an. Haben wir es mit einer Neuheit zu tun? Haben wir es damit zu tun, dass wir es eben nicht gut mit anderen Merkmalen abschätzen können, was jetzt das Richtige ist?
Also ich finde das sehr überzeugend, vor allem wenn das sehr einfach in der Anwendung wird und ohne zu viel vom Anwender zu verlangen. Was diese Abhängigkeiten und letztendlich dann auch quantifizierten Abhängigkeiten betrifft. Das sehe ich so als relevanten Punkt an.

Ich denke an ein Beispiel, wo wir uns schon ein bisschen iterativ annähern mussten, wo wir dann aber auch systematisch versucht haben, das mit einfachen Ansätzen zu ersetzen. Da bin ich mir nicht sicher, ist das denn abbildbar. Aber gut, das muss man tatsächlich ausprobieren und vielleicht auch einfach schauen, was man da macht. Das ist zum Teil eben auch schon ein geometrisches Problem gewesen, wo ich jetzt noch sagen würde, wie würde ich das denn da abbilden? Kann ich mir jetzt nicht so ganz vorstellen.

[59:00] **Hast du da ein konkretes Beispiel?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Wir sind dann einerseits dazu übergegangen, erstens diese sogenannten Netze darzustellen. Und andererseits haben wir gesagt, am schwierigsten sehen wird die Erfüllung der Leistung. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[1:01:32] **Und da habt ihr jetzt schon so in gewisser Weise ein Wirknetz erstellt, oder ist es die Idee das jetzt so zu machen?**

Wir haben jetzt natürliche nicht den Solutions Space Ansatz verwendet, aber wir haben natürlich angesagt: Was sind Eingangsparameter und was sind da die Erwartungen? Wie bedeutend sind die? Wir haben dann tatsächlich auch entschieden, was wir erwarten, was größeren Einfluss haben wird, was vielleicht einfach einen geringeren Einfluss hat, in welche Richtung die optimiert werden sollten. Und überall dort, wo es nicht so einfach möglich war eine Antwort zu geben wie der Formel Zusammenhang sein wird, haben wir dann einfach Tests gemacht? Also ganz schlicht und ergreifend.

[1:02:27] **Also die Zusammenhänge dann durch Tests validiert?**

Nein, also wirklich ganz einfache Tests. Wir haben dann keine XXXXX mehr genommen, sondern einfach Rundstäbe und haben geschaut was passiert, wenn die Auflagestellen eher in der Mitte liegen, eher außerhalb liegen, wenn sie größer werden. Um auf beliebig gekrümmten Flächen einen stabilen Stand zu erreichen, brauche ich ja eigentlich Füße. Und Füße führen aber schnell dazu, dass sie die Kontaktfläche geringhalten und das war dann ein Aspekt, der entscheidend war. Wenn ich wirklich Kontaktfläche opfere, wie lange macht es Sinn, die auf nahezu Null zu bringen, nur damit ich die Füße möglichst so stellen kann, dass sie über alle XXXXXX passen und eben dann trotzdem noch gute Kontakt Situation haben. Das konnte man zum Beispiel dann auch mit einfachen Experimenten, wie dem Rundstab relativ schnell bestimmen. Wie dünn die Füße werden dürfen und dergleichen.

[1:04:00] **Siehst du einen Anwendungsfall (für die Produktfamilienauslegung) bei euch im Unternehmen? Oder ist es nach wie vor so, dass du der Meinung bist, das hat nicht so viel Relevanz?**

Ich bin da hin und hergerissen, weil wir tun's ja eigentlich gar nicht und überall wo man sowas gar nicht zur Anwendung bringt, muss man davon ausgehen, dass wir da was liegen lassen. Hat aber sicherlich auch ein bisschen mit den Randbedingungen zu tun, die wir als Unternehmern besonders haben. Das heißt, man müsste wahrscheinlich, um da wirklich gut einen Vorteil daraus zu generieren, bestimmte Unternehmensrandbedingungen auch ändern oder verschieben. Aber das ist ja nicht falsch per se.
Uninteressant finden ich es nicht. Ich sehe es jetzt nicht per se in der Familien Auslegung. Also nicht, dass wir jetzt drei XXXXXXXXX bieten wollen, aber letztendlich kann man das ja auch anders sehen über Produktvarianten, wo dann bestimmte Sterne einfach bei verschiedenen Varianten anders zum Tragen kommen.

[1:05:16] **Es gibt ja schon durchaus Produktlinien, die ich jetzt als Variante bezeichnet hätte. Zum Beispiel die XXXXXX. Inwiefern da die Anforderungen variieren, bis auf die Geometrie, weiß ich nicht...**

Also, wir haben da bestimmte Beispiele. Die Frage ist, welche Ebene man betrachtet. Betrachtet man wirklich unterschiedliche Produkte oder sogar in die Richtung Komponenten. Da kann man ja auch die Betrachtung ziehen. Na wobei, also für das Beispiel jetzt nicht.

[1:05:50] **Wie sieht es denn mit Kostenmodell aus bei euch in der Entwicklung. Gibt es da Betrachtungen oder Modelle, auf die man zugreifen kann. Gerade eben in Hinblick auf Einkauf- oder Fertigungskosten und wie die entstehen.?**

Nein, das gibt es nicht. Kostenmodelle indem Sinne gibt es nicht. Da bräuchte man ja eigentlich auch eine Überführung von Parameter basiert oder aus den 3D Daten selbst.
Das haben wir in beiden Fällen nicht. Wir haben vielleicht Vergleichswerte mit ähnlich Produkten. Das ist eigentlich so das Entscheidendste.

ABSCHLUSS

[1:06:47 **Hast du denn tendenziell Interesse, dass wir einen der Ansätze mal ausprobieren bei euch im Unternehmen? Zum Beispiel in Form einer Studienarbeit. Dass man sich in irgendeiner Weise ein Produkt genauer anschaut.**

Eine Studienarbeit?

Das war jetzt nur ein Beispiel.

Ja das hat mich jetzt tatsächlich überrascht, weil ich hätte jetzt eher gedacht, dass das mit einmal ausprobieren ist, aber das wird dem Zeitaufwand wahrscheinlich nicht gerecht.

[1:07:32] **Ich kann dir als Beispiel nennen: Die kostenoptimierte Familienauslegung mit der Schlauchtrommel, das war auch eine Masterarbeit, wo dann dieses Verfahren entwickelt wurde mit den Kostenmodell und so weiter. Aber es ist immer interessant, es nochmal bei anderen Produkten zu sehen, auch um zum Beispiel bei den Datenbanken zu sehen, wo man wieder auf Grenzen oder auf Besonderheiten stößt. Es ist ja schon branchenspezifisch im Moment noch.**
Es geht auch eher darum, ob du weiterhin Interesse an dem Thema hast. Dann würde ich dich auf dem Laufenden halten.

Also Interesse an dem Thema auf jeden Fall. Ich denke auch, dass es ja, wie gesagt zu unserem Vorgehen passt, nur eben das Ganze professioneller lösen kann.
Die Aufwände sind halt spannend, die dann entstehen dadurch. Das ist auch ein Thema, warum ich da eben sozusagen ein bisschen überrascht war. Das erfordert dann natürlich auch einen ganz konkreten Anwendungsfall mit entsprechender Vorbereitung der notwendigen Tiefe, damit die Studienarbeit Hand und Fuß hat. Und eigentlich muss man auch sagen, müsste es dann entkoppelt werden von Entwicklungsarbeiten oder -projekten, die aktuell stattfinden. Es sollte eigentlich eine Vorbereitung für ein angedachtes Projekt sein.

[1:09:14] **Ja zum Beispiel. Aber das geht jetzt auch sehr weit in die Tiefe. Wir werden auch anhand der Ergebnisse noch Workshops organisieren. Da kann ich dir dann auch Bescheid geben. Vielleicht habt ihr Lust daran Teil zu nehmen.**

Also ich finde das sehr interessante und wie gesagt, auch in die richtige Richtung führend. Was vielleicht für mich jetzt noch nicht so transparent wurde, bei dem Anforderungsmanagement.

[1:09:49] **Was war da nicht transparent?**

Wie das da in dem Beispiel aufgegriffen wird. Aber das liegt wahrscheinlich daran, dass wir da weniger Schwierigkeiten haben. Die Schwierigkeit für uns bei Anforderungen besteht eher zu einem vollständigen, wirklich vollständigen, nicht nur scheinbar vollständigen Anforderungssatz zu kommen, der auch wirklich quantifiziert ist. Und wo wirklich die richtigen Anforderungen drin sind, die also nochmal überprüft sind.

[1:11:30] **Ich halte dich auf jeden Fall auf dem Laufenden. Sinn der Studie ist ja auch, erstmal die Resonanz aus der Praxis abzufragen.**

Ich kann mir schon vorstellen, dass die gut ist. Aber natürlich mit der Schwierigkeit, das jetzt greifbar zu machen, Aufwände zu sehen und dergleichen.

Das ist auch was, wo ich bei Simulations Tools immer so ein bisschen zurückhaltend bin, weil ich festgestellt habe, dass viele nach Simulations Tools schreien und sagen wir brauchen dies und das. Man braucht die besten Werkzeuge, aber keiner hat eigentlich das handwerkliche Geschick, um die richtig einzusetzen.

Und das nächste Thema ist dann auch, wieviel Zeit bin ich bereit zu investieren? Ich finde es ansonsten irritierend, wenn jemand nach einem teuren Werkzeug schreit und er aber jetzt eigentlich schon gar nicht auf Papier oder mit anderen einfachen Mitteln versucht, sich der Lösung systematisch zu nähern oder sie zu verstehen. Dem muss doch klar sein, dass er mit so einem Tool zwar verbessere Ergebnisse kriegt, aber eigentlich noch mehr Zeit investieren muss. Und ein Tool hat halt auch die Herausforderung, dass ich nicht zwangsläufig verstehe, was da stattfindet. Welche Annahmen dahinterstecken.

Wenn ich mit meinen einfachsten Werkzeugen auf Papier mir Zeit nehme, um zu einer Lösung zu kommen, dann gehe ich davon aus, dass ich das zur Anwendung bringe, was ich auch verstehe. Und ich weiß, welche Annahmen ich dabei treffe. Es fängt schon an bei der FEM Simulation, da sieht man, dass viele eben nicht verstehen, was sie da tun und was sie für ein Ergebnis kriegen. Oft wird im Zweifelsfall keine Konvergenz überprüft. Oder die Art der Einspannung wird vielleicht nicht überprüft, ob das jetzt dem entspricht, was eigentlich plausibel ist und erwartet ist.

[1:13:36] **Man gibt das dann dem Tool bisschen in die Hand. Also man verlässt sich zu sehr, auf das, was dort passiert, ohne zu wissen was im Hintergrund eigentlich läuft. Meinst du das?**

Ja, das sehe ich schon. Eine Schwierigkeit, weil eigentlich ist das Tool mächtiger zu dem Zeitpunkt, als man selbst und jemand, der hier wenig Zeit investiert und das Tool auch nicht wirklich versteht, der kann sich da so manches Problem künstlich generieren.

Das ist auch ein Aspekt, dass man dann vielleicht auch mal eben dazu neigt, blind solchen Ergebnissen zu vertrauen. Das ist ja das Stichwort FEM. Ich kann jedes Ergebnis bewirken und verändern. Ich kann sogar die farbliche Gestalt verändern, dann ist irgendwo rot, wo eigentlich nix passiert.

Und wie gesagt, es ist ja nur ein Bild und ob da Konvergenz ist beispielsweise, das wird auch nicht überprüft. Also das ist die Gefahr, dass man sagt: und damit war's des. Weil man sich nur so oberflächlich damit beschäftigen muss.

Das wird aber genau die Herausforderungen des Tools sein, dass das gelingt, dass man diesen Modulen vertraut, dass sich eben keine zeitlichen Aufwände habe und es wirklich auch keinen Raum gibt, da groß Fehler zu machen. Deswegen bin ich gespannt darauf, wie das in der Praxis dann aussieht. Aber wie gesagt, deckt sich auf jeden Fall gut mit den Ansprüchen, die wir haben, dass man mal das physikalische Wirkprinzip wenigstens versteht und sich klarmacht, welche Kennzahlen da sind.

HYPOTHESEN

[1:16:16] **Du siehst hier Aussagen, die du im Laufe des Interviews schon gehört hast. Stimme da bitte einfach mal ab, wie du das aus dem Bauch raus siehst.**

Die letzte Frage ist auch auf jeden Fall interessant. Es ist schon spannend zu sehen, dass das Entwickeln auch zum Teil einfach solchen Tools überlassen wird. Es gibt ja auch ganz interessante Werkzeuge, die tatsächlich auch dein komplettes Design machen können, basierend auf den Randbedingungen, die du setzt. Das finde ich schon auch interessant und wenn man das dann kombiniert mit sowas wie additiver Fertigung oder so, dann ist es ja fast nur noch Anforderungen bestimmen. Den Rest machen dann die Tools. Das ist auf jeden Fall interessant. Ich glaube, in der

Praxis zieht es ja alles noch nicht, aber deswegen ist die frühe Phase halt so wichtig, weil die folgende Phase halt vielleicht wirklich irgendwann zum Teil automatisch gelöst wird.

Danke für die Teilnahme am Interview.

Nr. Interview	05	Unternehmen	X	Datum und Ort	28.05.2020, München	Interviewer	XXX
Zoom Meeting, audio und visuell, angenehme Atmosphäre. In ein paar Produkte und Prozesse Einblick. Gibt im Rahmen seiner Möglichkeiten Antworten. Wo wenig technisches Wissen vorlag, Themen nur angeschnitten; hat das Video nicht zu Vorbereitung gesehen. Dauer: 59:08							

[02:33] **Stell dich bitte kurz vor, mit deiner Position im Unternehmen und was da deine Aufgaben sind.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

BEGINN HAUPTTEIL

[06:27] **Abgesehen davon, in welchen du jetzt involviert bist oder nicht, fallen dir weitere Herausforderungen in deinem Arbeitsfeld ein, die damit [genannte Themengebiete] nicht abgedeckt werden?**

Weiß ich jetzt nicht auf Anhieb.
Sofort, nachdem man ein Produkt mehr oder weniger hat, kommt es eher darauf an, Daten für dieses Produkt zu haben. Aber in sämtlichen Richtungen Vertriebsdaten, technische Daten, Anleitungen, viel Marketing Daten,
Da sehe ich immer die Schwierigkeit, dass man immer unheimlich viele Daten für verschiedene Produkte bereitstellen muss in sämtliche Richtungen und die kommen nicht automatisch.

[07:34] **Also die Verfügbarkeit der Daten dann?**

Genau. Erhebung während des Entwickelns, da muss man meistens dann wieder hinterher darangehen. Das ist natürlich extrem aufwendig, weil es wird während des Entwickelns nicht so viel daran gedacht, dass es später noch relevant ist. Und dann muss man sich hinterher wieder alles zusammensuchen an Daten, die relevant sind, was eigentlich schon mal irgendwo erhoben wurde. Man redete hier immer von Big Data. Aber darum geht es nicht so sehr. Das sind einfach Produktinformationen, die irgendwo wichtig sind, jedem irgendwie weiterhelfen können, die aber auf dem Weg mal erhoben werden und dann wieder verloren gehen.
Wie es so schön heißt, Wissensmanagement könnte man noch mit aufnehmen oder Informationsmanagement, Informationsspeicherung.

Ansonsten, was vielleicht auch ab und zu schwierig ist, vielleicht bei meinen Produkten nicht, aber bei Geräten, die wir haben. Man denkt nicht immer sofort an die After-Sales Themen, an Service Themen oder an Wartungsthemen. Wie ich den Punkt nennen soll, weiß ich nicht. Vielleicht zählt es auch zum Robusten Design.

[08:57] **Vielleicht eher Anforderungsmanagement?**

Ja das könnte man sagen.

[09:09] **Wie werden solche Anforderungen, wie Wartungen zum Beispiel gestellt? Wie sieht da eine typische Anforderung aus?**

Die Frage verstehe ich jetzt nicht.

[09:23] **Es geht darum, dass die Anforderungen, die wirklich konkret quantitativ erstellt werden, für unseren Ansatz besonders wertvoll sind. Also konkrete Zahlenwerte haben. Im Falle von Service und Wartung. Habt ihr da Quantitäten, wie Lebensdauer oder bestimmte Zyklen, die die Maschinen durchlaufen müssen?**

Achso, ja. Das ist so ein Marktstandard, der vorgegeben wird. Man beleuchtet den Wettbewerb, wie ist es der Markt gewohnt zu warten? Und da ist es einfach typisch, dass die meisten ein XXX Monatszyklus zum Beispiel haben. Das ist auch so, dass es im Unternehmen typische Wochen gibt, in denen gewartet wird, einmal im Jahr. Und deshalb könnte vielleicht ein Gerät auch locker XXXXXXXXXXXXXXXXXXXXXXXXXXXX nicht gewartet werden, das wird da gar nicht erst wirklich erhoben, weil es gibt einfach diesen Marktstandard von XXX Monaten und dann nimmt man den her. Darauf versucht man sich einzuschießen, weil es ist sehr schwierig, dann den Kunden zu überzeugen, wie er es von unserer Seite aus richtig zu tun hat, wenn er sowieso von sich aus schon seit Jahren das Gefühl hat, alle XXX Monate muss ich eine Wartung machen, dann nutze ich das auch aus, weil ihm vom Gegenteil zu überzeugen ist viel schwieriger als dann im Strom mit zu schwimmen.

Also ja. Zum Beispiel Wartungszyklen kommt vom Markt aus. Viele Dinge kommen eigentlich vom Markt aus, wenn man den Wettbewerb betrachtet und das, was der Kunde gewohnt ist. Nur wenn es dann um Eigenschaften geht, die nur bei unseren Produkten vorkommenden, dann versucht man irgendwie herauszufinden wie kann ich das quantifizieren? Zum Beispiel wie genau muss ich messen. Also, wie genau muss ich mein Gerät einstellen? Man versucht da zum Beispiel herauszufinden, was ist der Kunde, was möchte er genau messen mit unserem Gerät? Und wie genau muss sich das Gerät einstellen, damit es dann hinterher auch beim wiederholten Messen die richtige Messgenauigkeit hat? Also das kommt dann wieder aus der Zielanwendung raus.

…

[12:14] **Sortiere diese Begriffe mal der Relevanz nach, so wie sie für dich am relevantesten erscheinen in deinem Umfeld. Und dann gehen wir die Themen in der Reihenfolge durch.**

Es ist ganz schwierig, hier zwischen Theorie und Praxis jetzt in meinem Kopf zu unterscheiden.

1. Anforderungsmanagement 2. Entscheidungsfindung 3. Robustes Design 4. Variantenmanagement 5. Komplexität

---

ANFORDERUNGSMANAGEMENT

.

[14:34] **Anforderungsmanagement ist der erste Punkt. Ganz allgemein, welche Herausforderungen treten in der frühen Phase bei der Stellung von Anforderungen auf?**

Man braucht ja Anforderungen, weil man macht ein neues Produkt. Da muss erstmal geklärt werden, wer soll denn für dieses Produkt verantwortlich sein? Und wer gehört zum Team dazu, die dann hinterher teilnehmen, dieses Produkt zu entwickeln und zu vermarkten etc.? Das heißt, man muss erstmal ein Team feststellen und dann muss natürlich im Team entschieden werden, wer kümmert sich darum, die Anforderungen zu erheben. In dem Fall bei uns, bin das in den meisten Fällen ich oder ich und zwei, drei Kollegen, die in diesem Bereich arbeiten.

Ich versuche jetzt ein bisschen darüber nachzudenken, wie ich das bei verschiedenen Sachen gemacht habe. Wie schon gesagt, ich versuche Anforderungen zu finden im Markt, das heißt beim Wettbewerb erstmal. Heißt die Herausforderung ist erst mal da, komme ich an aktuelle Wettbewerbsinformationen. Also sind die Wettbewerbsinformationen aktuell. Stimmen die noch? Bekomme ich diese Informationen überhaupt? Und sind die dann überhaupt noch sinnvoll für mich?

Das heißt, ich sammle erst mal so viel wie möglich und versucht dann zu evaluieren, was ist überhaupt notwendig für mich. Zeitgleich versuche ich, die Kunden zu befragen. Es fällt aber manchmal recht schwer. Also nehme ich meine Verkäufer. Wir haben deutschlandweit und europaweit sicher XXXX Verkäufer, die regelmäßig zu Kunden fahren. Die haben natürlich auch ein gewisses Wissen, das versuche ich abzugreifen. Genauso gibt es aber auch Techniker, die wirklich die Wartungen schon machen bei Kunden. Und die frage ich auch, was ist den Kunden wichtig, wenn ihr da reingeht? Weil das sind durchaus unterschiedliche Aspekte. Der Verkäufer redet vielleicht mit einem Einkäufer aber der Techniker redet mit einem anderen Techniker. Das heißt, ich versuche, möglichst viele Sichten vom Kunden reinzubringen.

Wie mache ich das? Verschiedene, auch Fragebögen, habe ich mal versucht. Aber das ist in der Praxis sehr schwer umzusetzen. Das heißt eigentlich findet alles über Gespräche statt. Man versucht einfach, mit den Leuten zu telefonieren, sie zu treffen und stellt vor, was man gerade macht. Einen kontrollierten Fragebogen, an dem man sich entlang hangelt, musste ich leider aufgeben. Ich habe es probiert, aber es hat nicht gut funktioniert, weil gerade die in der Industrie, die sind das nicht gewohnt und sind dann sehr schnell überfordert mit diesem Fragebogen zum Beispiel. Also versucht man das in Gesprächen zu klären und macht sich die wichtigsten

Notizen. Und dann braucht man tatsächlich manchmal mehrere, auch Kollegen, die einen dabei unterstützen, weil so viel sprechen kann man gar nicht, wie man da eigentlich Leute interviewen muss. Da versucht man eben dann wieder im Team zu beraten, was hat der gesagt, was hat der andere gesagt?

[18:17] **Gut, und wenn ihr dann die Anforderungen gesammelt habt, wie werden die dann erfasst. Habt ihr da Werkzeuge oder Tools, wie ihr die dann zusammenfasst?**

Da haben wir ganz klar kein Tool. Das ist immer demjenigen überlassen, der für das Anforderungsmanagement zuständig ist. Das heißt, je nachdem, was ich habe, führe ich eine einfache Word Dateien oder Excel Datei. Oder wenn es dann soweit kommt, dass man das nochmal der ganzen Gruppe, den Entscheidern, der Geschäftsleitung vorführt, dann ist natürlich auch viele komprimiert in einer Powerpoint drin. Wir arbeiten hier mit diesen ganz klassischen Office Dokumenten. Es gibt kein Anforderungsmanagement System, wie es andere Firmen haben.

[19:14] **Wie viele Anforderungen sind quantitativ erfasst bei euch? Kannst du da eine Prozentzahl geben?**

Nein, da kann ich leider nichts zu sagen. Es gibt welche, die kann man quantitativ erfassen. Das sind gewisse Preise, Rabatte. Bei manchen Produkten sind es die Geometrien und die Abmaße. Aber gerade beim Service zum Beispiel wiederum da könnten so viel Anforderungen drin sein, wo ich jetzt erst über längeres Nachdenken darauf komme, das es überhaupt eine Anforderung ist. Aber die sind nicht quantitativ, die sind meines Erachtens eher qualitativ dementsprechend. Ich kann da leider keine Zahl nennen.

[20:15] **Wie häufig treten Zielkonflikt in der frühen Phase auf? Wenige Male, hin und wieder oder bei fast allen Anforderungen?**

Bei fast allen Anforderungen.

ENTSCHEIDUNGSFINDUNG

[20:35] **Woran hast du gedacht, an welche Herausforderungen bei der Entscheidungsfindung?**

Ja, jetzt wo du das Wort Zielkonflikt nennst. Es gibt immer zwei oder mehrere Möglichkeiten und man muss sich jetzt auf eine festlegen. Und die Herausforderung ist zum einen, man kann die Zukunft ja nicht voraussagen und das ist diese Unsicherheit. Ich nehme jetzt aus den drei Möglichkeiten eine, was passiert, wenn es falsch ist? Wären die anderen nicht doch besser? Jeder hat eine Meinung dazu, die ist unterschiedlich und es ist sehr schwierig, entweder einen Kompromiss zu finden oder sich eines davon auszusuchen. Es ist schwierig dann auch davon abzusehen, was werden die Folgen davon seien, wenn ich mich dafür entscheide.

[21:54] **Hast du auch einen Blick auf Designentscheidungen konkret, wenn ihr die Anforderung gesammelt habt und euch dann für Konzepte entscheiden müsst?**

Ganz selten. Ab und zu kommt es schon mal vor, aber eigentlich eher selten.

[22:21] **Dann würden mich mehr noch die Fragen zu Robustem Design und Variantenmanagement interessieren.**

Versuchen wir es mal genau. Das ist wahrscheinlich der mit Abstand schwächste Punkt bei mir, weil ich nicht direkt neben den Konstrukteuren und Entwicklern sitze, die dann wirklich das Produkt ausdesignen.

[22:36] **Ok, dann eine Frage noch zu der Entscheidungsfindung. Vielleicht hast du da ja trotzdem einen Überblick, wie hoch der Anteil an Simulationen und Berechnungen bei der Auslegung der Produkte ist.**

Hm. Also ich weiß natürlich, dass wir Produkte haben, wo vieles simuliert wurde. Da wurden auch FEM Analysen gemacht.

Grundsätzlich wird sehr wenig simuliert. Das war mal ein Produkt, wo man das auch in Kooperation sehr stark gemacht hat. Aber bei uns in der Firma ist es eher so, dass eher mehrere Prototypen wirklich gebaut werden und dann auf Testanlagen getestet werden. Oder Geräte werden gebaut und werden dann Langzeit getestet in der eigenen Fertigung. Also es ist nicht so, dass wir sehr viel simulieren, sondern wir machen es einfach und schauen, wie das fertige Produkt wirklich ist.

[23:41] **Wie viele Prototypen entstehen da so im Schnitt für ein Produkt?**

Kann ich nicht sagen, weil ich nicht alle zu Gesicht bekomme. Ich sehe dann mal drei Stück, aber es könnten genauso gut 10 bis 20 davon gewesen sein. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Kann alles sein, oder es ist zuerst auf einer Zeichnung und dann machen Sie von 10 Zeichnungen drei und von den dreien sucht er dann eins aus. Aber ich bin in diesen Entscheidungen zu seltenen dabei, das weiß ich leider nicht. Deshalb kann ich da auch keine Zahl nennen.

ROBUSTES DESIGN

[25:07] **Da geht es hauptsächlich darum, dass Anforderungen im Laufe der Produktentwicklung häufig geändert werden müssen. Wir verstehen unter robustem Design die Fähigkeit des Produkts, auch bei sich ändernden Anforderungen trotzdem noch die Gesamtanforderungen zu halten. Oder wenn sich Umwelteinflüsse ändern und so weiter, dass man trotzdem noch in einer validen Lösung bleibt, die nicht so anfällig ist für Änderungen. Entstehen bei euch im Unternehmen Probleme durch die Änderungen von Anforderungen?**

Ja, das kann schon vorkommen. Nur wenn ich jetzt meine Produkte betrachte, wenn wir jetzt auch über die Herausforderungen reden, könnte man das in zwei Segmente unterteilen.
Zum einen habe ich Geräte, die natürlich aus extrem vielen Bauteilen bestehen, da natürlich auch Software drin ist. Da muss ich auf jeden Fall ein robustes Design haben, weil die Ansprüche der Kunden steigen. Und ich bin auch irgendwie gezwungen alle ein bis zwei Jahre wirklich auch eine Innovation an den Geräten rauszubringen. Da ist die Frage, ich habe eine Platine drin. Kann ich diese Platine auch in unterschiedliche Gehäuse bringen? Bleibt es egal? Auch, wenn der Kunde ein anderes Design hat, kann ich trotzdem die Technik dahinter gleich lassen? Und jetzt haben wir erst vor kurzem eine komplett neue Elektronik aufgesetzt, die muss natürlich auch robust sein gegenüber vielen neuen Softwareveränderungen, die da draufkommen.
Das Schwierige ist jetzt, das ist wirklich nicht mein Produkt, aber deshalb kann ich dir nicht ganz so viel dazu sagen. Aber da ist natürlich schon wichtig, dass die einmal getane Arbeit für eine Platine, die das Herz dieses Gerätes darstellt, dass die natürlich auch über die Dauer funktioniert. Bis jetzt hat sich das eigentlich ganz gut ausgezahlt. Wobei auch da der Markt manchmal noch nicht genau weiß, was er wirklich braucht und man braucht nicht so viel Veränderungen in diesen Geräten. Die werden jetzt bei uns seit wahrscheinlich XXX Jahren so entwickelt, wie sie sind und man musste kaum was daran ändern, weil so viele Anforderungen sind nicht gekommen. Mittlerweile kommt mehr dieses Software Thema dazu, XXXXXXXXXXXXXXXXXXXXXXXXXX

Ansonsten bei den Produkten, wo ich hauptsächlich zuständig für bin. Da hast du ein einziges Stück XXXX. Also XXXXXXXXXXXXXXXX hat keine Komponenten, die man dran baut oder sonst irgendetwas. Das heißt, ein Produkt wird einmal ausdesignt und so wird es gefertigt und so hat es eine Artikelnummer. Wenn jetzt die Anforderung wechselt, da muss ich einfach sowieso ein neues Produkt machen. Und dann muss ich es gleich in mehreren Varianten machen, weil man braucht immer eine gewisse Breite. Da brauch ich das Produkt gar nicht so robust designen, weil viele Dinge sind auch wieder durch Standards und DINs vorgegeben. Und die einzige Aufgabe, was wir dann haben, ist es zu entscheiden, wollen wir die neuen Anforderungen umsetzen oder nicht? Und da braucht auch das vorherige Produkt nicht robust sein, weil es gibt sowieso ein neues. Also da stellt sich die Frage gar nicht.

VARIANTENMANAGEMENT

[28:45] **Das ist jetzt interessant, auch im Hinblick auf das Variantenmanagement. Wenn du sagst, dass ein Produkt aus einer Komponente besteht, habt ihr trotzdem Familien oder sowas wie eine Familienbildung im Unternehmen?**

Die ist definitiv da, für alle Produkte. Manchmal hat sie eine gewisse Logik. Manchmal ist sie einfach historisch gewachsen. Da triffst du jetzt einen Nerv, weil das ist gerade aktuell. Wir versuchen einen E-Shop aufzubauen und unsere Internetseite umzukrempeln. In diesem Sinne muss man natürlich seine Produktdaten auch besser bereitstellen. Und jetzt muss das in eine Produktfamilie rein, und das wird dann sehr schwierig, Dinge, die historisch gewachsen sind, in Variantenmanagement und Daten abzubilden.
Es gibt definitiv Familien und man kann das sehr, sehr schön zuordnen. Aber es ist schwierig es abzubilden.

…

TEIL2

[41:50] **Ich würde dich bitten, dass du nochmal über die Forschungsansätze nachdenkst und überlegst, ob du einen Anwendungsfall bei dir im Unternehmen siehst.**

Das ist ziemlich schwieriges, ziemlich abstrakt.

[42:11] **Was ist abstrakt?**

Jetzt habe ich hier die zwei Bilder mit der Kurve und diesem Wirknetz und versuche das irgendwie in Produkte einzuordnen.

[42:29] **Wir können ja links bei der Familien Auslegungen anfangen, da kann ich dir kurz bisschen helfen. Habt ihr denn auch Produkte, die aus verschiedenen Komponenten entstehen?**

Ja, das wären dann auf jeden Fall die Geräte.

[42:45] **Und kommt es da manchmal zu Problemen wie diese verschiedenen Komponenten zusammengefasst werden und ob man überhaupt Komponenten zusammenfassen kann?**

Ja, denke ich, dass es das schon gibt.
Es gibt auf jeden Fall Komponenten, die für verschiedene Geräte verwendet werden können, wiederum Komponenten, die nur auf der einen funktionieren.

[43:13] **Weißt du, ob es da Herangehensweisen gibt. Ob ein solches Komponenten Sharing betrachtet wird?**

Ich denke ja. Es wird schon darauf geachtet, dass Systeme durchgängig und möglichst breit angewendet werden können. Oder auch Komponenten, die schon existieren. Sei es Zubehör oder sei es interne Komponenten. Nur kann ich leider nicht genug dazu sagen. Ich denke da wäre es zwar schön, kostenoptimierte Familienauslegung zu machen. Aber ich glaube, da wird sehr viel mehr auf technische Anwendbarkeit geschaut. Also es muss halt funktionieren und danach kann man noch versuchen, im Rahmen der Möglichkeiten, Kosten zu optimieren. Aber meistens wird dann nicht mehr viel am Design oder sonst irgendwas geändert.

[44:13] **Gibt es Kostenmodelle, zu denen ihr Zugriff habt in der Auslegung?**

Nein, ich gehe nicht davon aus.

[44:27] **Fällt dir noch was zur Modellerstellung ein? Könnte das an einer Stelle erleichtern?**

Ja, irgendwann, irgendwo wird es sicher etwas erleichtern. Aber wenn du mich fragst wo, dann weiß ich es nicht. Es ist sehr, sehr schwierig zu greifen und irgendwie umzusetzen in ein Produkt.
Da muss ich leider passen an der Stelle.

[45:04] **Wir können ja nochmal eine andere Sicht einnehmen. Wenn du jetzt nicht aus Sicht des Mitarbeiters in deiner Position sprichst, sondern als Experte oder Wissenschaftler allgemein. Was fällt dir noch ein, was du zu den Ansätzen sagen willst? Oder an welcher Stelle es noch Verbesserungen oder Erweiterung gibt? Was dir so ins Auge gefallen ist, als ich die Themen vorgestellt habe.**

Das Erste, was mir kommt, ist die Frage, bei wie vielen Kunden oder Ziel Anwendung findest du Leute, die intelligent genug sind das mitzumachen. Ich glaube, es ist extrem schwierig. Extrem schwierig zwischen all dem, was der alltägliche Konstrukteur, Entwickler sonst was macht, sich dann so damit auseinanderzusetzen.

Wenn ich daran zurückdenke, als ich XXXXXXXXXX gemacht habe, da war ich bei einer anderen Firma. Die haben auch versucht ihre XXXXXXXXX zu optimieren. Da ging es auch um Modulbauweise. Auch da haben wir eigentlich schon extrem gute Methoden angewandt. Aber auch da waren wir noch weit entfernt davon gewesen, so etwas zu nutzen und ich glaub, da war auch keiner in der Lage das so zu machen.
Gerade bei Mittelständlern, XXXXXXXXXXXXXXXXXXXXXXXXXXXX, wenn es gut läuft, dann sind da ein, zwei Ingenieure, die kamen von der Uni. Die kennen extrem viele Ansätze im Vergleich zum Rest, der einfach natürlich angestellt und entwickelt wurde bei der Firma aber trotzdem, das ist dann so hochtrabend. Und dann kommt immer dieser berühmt berüchtigte Elfenbeinturm. Das heißt, die Konzepte sehen gut aus, aber die Umsetzung mit dem Menschen, die man hat, das ist glaube ich extrem schwierig. Das ist immer das Erste, was mir da auffällt oder was mir selber leider auch aufgefallen ist, als ich dann damit konfrontiert wurde.

Je einfacher es im ersten Moment wirkt, desto besser kann man es anwenden. Und deshalb muss ich zugeben, selbst allein wegen der einfachen Grafik ist diese kostenoptimierte Familienauslegung und die relativ einfache Erklärung dazu denke ich ein extrem guter Ansatz, der verwendet werden kann.

Während beim rechten Modell habe ich immer noch meine Bedenken, dass man sehr schnell das Wirknetz zusammenkriegt. Die Schwierigkeit ist, alles zu betrachten, glaube ich. Also ich denke, dass man viel zu schnell einige wichtige Äste von so einem Wirknetz vergisst, weil man es nicht auf dem Schirm hat. Es ist dieses typische: Das machen wir schon immer so! Also Dinge wo man gar nicht nachdenkt und sie werden so gemacht. Aber man könnte sie auch ins Wirknetz mit reinbringen.

[48:22] **Genau dafür soll ja auch die automatisierte Modellerstellung dienen, das nichts vergessen wird. Dass man so Komplettkonzepte anbietet. Das wäre dann ein Potenzial, um die Umsetzung zu unterstützen.**

Ok, und dann quantifizieren.

Ich kann's mir extrem schwer vorstellen. Wobei ich bin immer so der Meinung, wenn die Denkarbeit gemacht ist, also für mich ist es das Wirknetz, dann ist das quantifizieren die Fleißarbeit. Das muss halt einfach gemacht werden. Aber wenn man die nötige Zeit gibt, funktioniert auch das.

AUSSAGEN UND HYPOTHESEN

49:38 **Bewerte, die [Hypothesen] mal nach eins, ich stimme völlig zu, bis fünf, ich stimme überhaupt nicht zu. Du kannst auch gerne Fragen stellen, ansonsten beantworte sie aus dem Bauch heraus.**

…

52:42 Hast du Interesse daran, dass wir dir weitere Ergebnisse schicken oder wenn wir einen Follow-up Workshop bezüglich eines Themas organisieren? Dürfen wir dich da kontaktieren?

Also Interesse an den Ergebnissen habe ich auf jeden Fall.

Follow-up Workshop kommt immer ein bisschen auf die Ausgestaltung drauf an, aber ich bin offen da auch weiter zu unterstützen. Kommt dann aber natürlich immer auf den konkreten Fall drauf an.

…

ABSCHLUSS

Die Krux ist zwischen dem, was ich weiß und gelernt habe aus der Theorie, aus meinem Studium und dem, was ich in den letzten Jahren in der Praxis mitbekommen habe zu unterscheiden. Man schwankt zwischen dem, was man gelernt hat, was richtig ist und dem, was gelebt wird in der Praxis. Es ist sehr interessant, dass dann auch extreme Unterschiede mal sein können, aber wenn man genauer hinguckt, dann auch manche Sachen Anwendung finden, obwohl es nicht so benannt wird.

Anfangs dachte ich, hier werden keine Anforderungen erhoben bei meinem Unternehmen. Aber das passiert schon, nur werden sie nicht in der Form dargestellt, wie man gelernt hat, dass sie dargestellt werden sollen.

Eine von den Fragen, die da waren, war: „auf den Fokus in der frühen Phase legen.“ Das lernt man, das ist extrem wichtig, weil in der frühen Phase wird entschieden hinterher kannst du nichts verändern. Da habe ich das aber im Unternehmen gesehen und dachte, von diesen Modellen ist gar nix anwendbar. Aber es wird schon am Anfang wirklich extrem viel Energie reingelegt, in der frühen Phase möglichst viel zu betrachten.

Wenn man sagt, zum Beispiel Szenario Analysen: Wer macht denn bitteschön sowas? Es ist immer eher qualitative Geschichten. Man denkt immer darüber nach, wenn ich das mache, wie läuft es raus? Euer Modell kann man auch als Szenarioanalyse betrachten, nur halt sehr, sehr viel detaillierter und mit Matlab abgestützt. Aber es ist schon immer das gleiche. Man versucht sehr früh daran zu denken wo will ich hin, was mach ich, was kann ich und was kommt dabei raus? Was ist das Schlimmste, was der beste Fall, was zwischendrin? Sowas wird immer bedacht, aber es wird nie so benannt. Das ist der Unterschied in der Praxis.

Danke für die Teilnahme am Interview.

Nr. Interview	07	Unternehmen	X	Datum und Ort	03.06.2020, München	Interviewer	XXX
Gespräch über Webex (Interview auf Nachmittag verschoben, nach anfänglichen Technikschwierigkeiten am Vormittag.) Zwischendrin kurze Störung durch Anruf. Dauer: 01:23:25							

[01:38] **Ich würde Sie erst einmal bitten, dass sich kurz vorstellen und vor allem auch auf die Position im Unternehmen eingehen und ihre Tätigkeiten dort.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[05:53] **Wo werden die XXXXXXXX?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[06:57] **Wir können bei diesen Baureihen noch mal kurz bleiben. Die XXXXX Komponenten hatten sie angesprochen.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[07:44] **Teilen Sie das auch in Komponenten ein, wie jetzt zum Beispiel XXXXX. Aber bei ihnen wird sicherlich auch anhand von Komponenten dann XXXXXXXXXXXXX zusammengefügt.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

In Ihrer eigenen Fertigung geht es da hauptsächlich um XXXXXXXXX oder werden auch komplexere Komponenten gebaut?

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

VARIANTENMANAGEMENT

[09:01] **Steigen wir gleich nochmal bisschen weiter mit ein und ziehen das Variantenmanagement mit rein. Entstehen da Probleme durch diese Vielzahl an Komponenten und Teile, die sie haben bezüglich Varianten.**

Wir haben das Problem, dass wir die Variantenzahl natürlich niedrig halten müssten, aus Kostengründen. Das ist ja sehr häufig so. Darum machen wir das so ähnlich wie die XXXX, wir machen Pakete, Ausstattungspakete zum Beispiel, sodass die Varianz dann reduziert wird. Wenn sie das meinen.

[09:47] **Ja genau, da ist der Punkt, den Sie genau treffen. Da ist die Frage, wie diese Varianten gerechte Gestaltung in der frühen Phase berücksichtigt wird.**

Also das funktioniert so, mir haben XXXXX eingeführt. Als Steuerungssystem für den ganzen Standort und wir haben die Variantenkonfiguration auf SAP Basis seit dem Zeitpunkt. Und da haben wir eine Abteilung, die sich praktisch um die Variantenkonfiguration in der Entwicklung kümmert und somit auch die Variantenvielfalt definiert. Dazu gibt es dann die Schnittstelle zum Marketing, das ist sehr wichtig, weil wir wollen ja die Kundenwünsche erfüllen und haben somit natürlich als Gegenseite, wenn man so will, die Anforderungen von Vertrieb und Marketing. Und dann gibt es in sehr vielen Detailgesprächen die Lösungsfindung, dass eine Baureihe - keine Ahnung, muss ich raten - XXXXX Varianten hat.

[10:59] **Wie werden denn diese Varianten ausgelegt? Können Sie da was sagen, aus Entwickler Sicht. Wie wird entschieden, welche Varianten gebildet werden und wo dann auch zum Beispiel Komponenten geteilt werden über verschiedene Varianten hinaus?**

Meinen Sie jetzt innerhalb der Baureihe oder über die Baureihen hinweg?

Je nachdem, wo sie Komponenten teilen. Also gibt es überhaupt Komponenten, die über verschiedene Baureihen hinweg geteilt werden?

Also praktisch Wiederverwendungsteile meinen Sie? Also Komponenten, die in allen Baureihen vorkommen?

Genau.

Ja, das machen wir natürlich, weil neue Teile verursachen am meisten Entwicklungsaufwand und haben auch höhere Kostengründe, darum ist natürlich jeder angehalten Wiederverwendungsteile zu verwenden. Dazu gibt es einen Katalog, der ist allen Konstrukteuren zugänglich, wo sie praktisch die Teile in Serie sehen und dazu angehalten werden, diese auch zu verwenden.

[12:04] **Wurde dieser Katalog schon einmal überprüft, ob der Kostengünstig ausgelegt ist. Also ob es da vielleicht bessere Standardisierungsmöglichkeiten geben könnte oder ist der historisch gewachsen?**

Das letztere. Der ist historisch gewachsen und ich glaube, dass der in diese Richtung gerne mal überprüft werden könnte. Es ist sicher eine Möglichkeit zur Verbesserung.

BEGINN HAUPTTEIL

[12:38] **Gehen wir hier nochmal auf die Übersicht der Herausforderungen, die aus unserer Sicht in der frühen Phase entstehen. Wir haben jetzt schon über das Variantenmanagement gesprochen.
Ich würde Sie bitten, dass sie sich die Begriffe anschauen und überlegen, ob ihnen noch weitere Themen einfallen, die häufig in Bezug auf die frühe Phase in der Entwicklung Herausforderungen birgt.**

Was in einem XXXXX Konzern ganz wichtig ist, es eigentlich, dann die Kosten-Nutzen-Rechnung, der Break-Even Point, also wann hat sich die Entwicklung amortisiert.

Wir machen in einer frühen Phase die Rechnung auf, welche Stückzahlen kann man absetzen und was kostet die Entwicklung?

Sie haben jetzt wahrscheinlich auch eh mehr die Sicht dieses Koordinators, als wirklich des Entwicklers, der dann die Teile aus konstruiert?

Ja, wie gesagt, ich habe selbst XX Jahre entwickelt. Also kenne ich den Bereich schon auch.
Und genau, wenn sie die Anforderungen an einen Entwickler speziell stellen, dann fällt mir dazu ein, das was ich vorhin schon gesagt habe, Wiederverwendungsteile. Dass man hier natürlich Priorität setzt. Dann Kosten bewusstes Konstruieren oder kostenoptimiertes Konstruieren. Und in einer sehr frühen Phase dann FE Rechnungen, gerade im XXXXXXX.

[14:49] **Darf ich da kurz fragen, wie das aussieht bei Ihnen. Wie groß ist der Anteil an Simulationen und Berechnungen während des Anforderungsmanagement zum Beispiel?**

Also, es gibt in der frühen Phase, wie schon gesagt, die FE Rechnung, die dann bezüglich Steifigkeit und Dauerhaltbarkeit oder auch Geräusch und Schwingungen, die ersten Aussagen bringen muss. Um hier natürlich kosten- und bauraumoptimierte Komponenten zu entwickeln.

Das ist das eine, was wir auch machen, sind XXXXXXXXXXXXXX. Ist ja auch immer wichtiger, dass die XXXX nicht zu XXXXXXXXX werden.
Das müsste es eigentlich dann gewesen sein. Das ist halt Konstruktionsbegleitend, wir haben eine eigene Abteilung, die führen die Simulationen und Berechnungen durch mit ca. XXX Leute. Die geben dann die Teile auch frei. Da gibt es dann verschiedene Events - heißt das bei uns - wo dann abgefragt wird, welche Serien Freigabe erteilt werden können an welche Abteilung.

[16:19] **In diesem Punkt. Wenn die Simulationsabteilungen die Teile frei gibt, dann sind die schon in der Entwicklung. Also dann wurde sich schon für einen Design entschieden. Habe ich das richtig verstanden?**

Wie Sie sagen, das ist in einer späteren Phase dann.

[16:35] **Und in der frühen Phase, werden da auch schon Berechnungen hinzugezogen?**

Ja. Also es ist so, dass ich selber XXXXXXXXXXXXXXXXXXXX konstruiert hab. Da hat man zum Beispiel das XXXXXXXXXXXXXXXXXXXXXX schon sehr früh simuliert, um hier XXXXXXXXXXXXXXXXXXXXXXXXXX und so weiter durchzuführen. Das wird sehr wohl gemacht, ja. In einer frühen Phase.

[17:19] **Gut, dann versuchen wir hier nochmal was Technisches. (...) Sie sehen hier nochmal die Themen von der Folie. Sortieren Sie die bitte mal der Relevanz nach, weil wir dann dementsprechend die Themen nochmal besprechen in der Reihenfolge.**

1. Anforderungsmanagement, 2. Komplexität, 3. Entscheidungsfindung, 4. Robustes Design, 5. Variantenmanagement

---

ANFORDERUNGSMANAGEMENT

[21:19] **Was fallen Ihnen beim Anforderungsmanagement für Herausforderungen ein?**

Also ich kann Ihnen sagen, wie das bei uns aktuell abläuft. Wir kriegen zum Start vom Projekt eine Produktbeschreibung vom Marketing, wo quasi die Anforderungen zusammengetragen sind, an ein neues XXXXXXXXXXX. Das ist so der Start jeden Projekts - die Produktbeschreibung. Das fußt normalerweise schon auf den ersten Berechnungen, wann ist das ganze Projekt amortisiert.
Und dahinter stehen auch Marktanalysen. Wo gibt es Möglichkeit, neue Absatzmärkte oder höhere Stückzahlen zu erzielen?
Ist das hier gemeint damit mit Anforderungen?

[23:07] **Ja, also es geht in die richtige Richtung. Jetzt wäre die für uns interessante Frage, wie das läuft, was es für Herausforderungen gibt, die momentan immer wieder auftreten.**
Wo kommen denn die Anforderungen hauptsächlich her?

Da gibt es eine Abteilung, die Produktmanagement heißt und die liefert praktisch die Daten und die Zahlen an die Entwicklung. Schnittstelle zwischen Marketing, Vertrieb und uns, der Entwicklung. So ist das organisiert und die hat eine Truppe von Leuten oder holt die Informationen vom Vertrieb ein, wo es jetzt neue Märkte gibt für XXXXXXX zum Beispiel.

[24:02] **Sind das dann auch schon speziell Funktionsanforderungen oder eher so Marktanforderungen?**

Das sind Marktanforderungen eigentlich. Also in der Produktbeschreibung steht zum Beispiel drin: XXXXXX braucht XXXXXXXXXXXX mit XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX. Das ist dann einfach so das, was der Markt momentan an Geräten hat, die wir dann betreiben müssen, stellt dann die Abteilung in die Projektbeschreibungen aus der wir dann das Lastenheft generieren.

[24:52] **Okay und zu dem Lastenheft: Wie viele Anforderungen werden da quantitativ erfasst in dem Lastenheft? Also technische Anforderungen vor allen Dingen.**

An technischen Anforderungen, ich kann Ihnen die Seitenzahl nennen, das sind zwischen XXXXXXXX A4 Seiten - je nach Projektgröße. Die Anzahl der Anforderungen oh, das ist gar nicht so leicht. Vielleicht wie viele stehen da drin, XXXX bis XXX Spezifikation stehen da drin.

[25:22] **Für uns ist besonders interessant, wie viele von diesen Anforderungen dann tatsächlich quantitativ erfasst sind. Also wo ein Zahlenwert dahinter steckt.**

Sie meinen also im Lastenheft, wie viel stehen da Anforderungen als Zahl drin. Ist das so richtig?

Genau.

Das sind dann vielleicht 50 % davon.

[26:01] **Sind die Anforderungen alle noch auf derselben Ebene oder unterteilen sie schon in Komponenten Ebenen?**

Wir unterscheidenden in Hauptkomponenten. Da sind dann XX Seiten XXX, XX Seiten XXXX XX Seiten XXXXX, XXXXX und so weiter. Da ist schon eine Gliederung.

KOMPLEXITÄT

[26:36] **Vor welche Herausforderung stellt Sie die Komplexität ihrer Produkte?**

Die Komplexität wird immer mehr, das kann ich schon mal sagen. Weil natürlich die Anforderungen ans XXXXXX immer größer werden. Stichwort XXXXXXXX zum Beispiel.
Ich meine, die Schwierigkeit ist die bei der Komplexität, wirklich einen Kundennutzen herauszufinden. Man kann ja das XXXXXXX beliebig kompliziert machen, aber wenn der Kunde kein Nutzen davon hat, dann bringt das am Ende nichts. Und wir versuchen die Komplexität stark einzugrenzen und das nicht ausufern zu lassen, indem wir ganz genau schauen, erstens mal was verlangt der Markt und zum Zweiten was macht die Konkurrenz.

[27:34] **Und das macht es dann komplex, weil verschiedene zusätzliche Anforderungen daraus entstehen? Oder was ist die Ursache dann für die Komplexität?**

Bei jeder Neuentwicklung kommen neuen Themen dazu, neue technische Herausforderung wie jetzt ganz aktuell XXXXXXXXXXX oder auch mehr XXXXXXXXXXXXXXX, so wie bei den XXXXX auch.

[28:29] **Wir haben auch schon Interviews mit XXXXXXXX geführt. XXXXXXXXXXXXXX ist ein großes Thema, dann auch in Verbindung mit vorausschauend planen. Was wird später mal in der Zukunft wichtig sein? Was will der Kunde überhaupt? Wie sieht es da aus bei ihnen? Sind die Horizonte klar, wo es hingeht?**

Ja, wir haben einen 7 Jahresplan, den das Topmanagement definiert. Und du sind die Ziele eigentlich klar gesetzt, wo die Entwicklung hin gehen muss - und zwar weltweit. XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[29:31] **Die XXXX produzieren sie aber nicht?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

ENTSCHEIDUNGSFINDUNG

[30:02] **Bei der Entscheidungsfindung geht es vor allen Dingen um Entscheidungen während der Konzeptphase, also die Entscheidung zwischen verschiedene Designs. Meistens eigentlich auch vor Projektstart, je nachdem, wo das bei ihnen stattfindet. Wie läuft es da ab? Haben Sie da Einblick?**

Wir haben verschiedene Teams, die eigentlich hauptsächlich mit Entscheidungsfindung beauftragt werden. Es gibt ein entscheidendes Team, bei uns nennt man das Entwicklungsteam. Das ist jetzt so zusammengestellt, das sind so zwischen fünf und acht Personen. Da sind alle relevanten Abteilungen und Bereiche dabei und die haben eigentlich die Aufgabe anhand von dem Lastenheft und von den daraus entstehen Pflichtenheften bei mehreren Möglichkeiten, die Richtige weiterzuentwickeln. Also im Grunde über ein Expertenteam, die richtigen Lösungen zu finden.
Das sieht dann sogar so aus, wenn es wirklich mal knifflig wird, dass wir eine Nutzwert Analyse durchführen.

[31:39] **Sind dann genug Informationen vorhanden, um die Entscheidungen treffen zu können und diese Nutzwert Analyse richtig durchzuführen? Oder fehlen häufig Informationen?**

Wir haben ja dann die Teams über mehrere Wochen laufen. Das geht ja oft über fünf bis acht Wochen. Wenn dann auch Simulation fehlen für die Entscheidungen, dann werden die in der Zeit eingeholt oder werden die Leute beauftragt, um die Simulation durchzuführen.
Und ja, dann muss irgendwann mal die Entscheidung fallen. Und da gibt es natürlich auch das Experten Know How, ohne dass man jetzt Nutzwert Analyse macht, dass die Leute, die einfach tief drinnen in der Materie sind, zu einem Konsens kommen und dann entscheiden.

[32:32] **Ist es erfahrungsgemäß dann eine Entscheidung, die basierend auf sehr vielen neutralen und quantitativen Informationen stattgefunden hat oder werden auch viele Bauch- oder Erfahrungsentscheidungen getroffen?**

Ich glaube Sie haben schon Recht. Es ist eine Mischung. Es kommt auch darauf an, um was es geht. Es sind schon viele Bauchentscheidungen auch dabei. Weil wir sehr viele Leute haben mir viel Erfahrung, die dann oftmals mit dem Bauch entscheiden.

ROBUSTES DESIGN

[33:15] **Danke, ich habe schon einen guten Einblick bekommen. Ich würde allerdings robustes Design beiseitelassen. Es sei denn, sie möchten zu dem Thema noch was sagen, was sie sich aufgeschrieben haben im Vorhinein.**

Ich habe beim robusten Design aufgeschrieben… Wir versuchen wirklich über Simulation, wo wir sehr lange Erfahrung haben, - also hauptsächlich FE Analysen - sehr früh die Komponenten robust auszulegen, weil wir mit den Anforderungen uns schwer tun alles abzudecken. Wenn Sie jetzt für einen neuen Markt zum Beispiel eine XXXXXXXXXX konstruieren für Brasilien, dann haben sie zwar XXXXXXXXXXXX in Europa oder auch in Amerika aufnehmen und messen können, aber nicht für Brasilien. Das heißt sie müssen für ein robustes Design, vor allem bei XXXXXX, sehr viel mit Simulationen machen und auch sehr früh Ableiten aus bestehendem Wissen. Sie können nicht alle Tests auf dieser Welt durchführen, um eine XXXXXXXX auszulegen. Das wird schlecht gehen. Und das andere ist auch, dass man sehr früh in Tests gehen.

[kurze Unterbrechung durch Anruf]

Jetzt bin ich wieder da. Also robustes Design war das Thema genau. Also ich habe aufgeschrieben: Zwei Stichworte. Erstens Simulation, richtig viel Simulation. Und zweitens sehr frühe Tests in den bekannten Märkten mit hohen Anforderungen. Das ist so unser Credo. Harte Tests durchführen mit den XXXXXXXX, sodass man dann wenig Probleme in den zukünftigen Märkten haben.

[35:47] **Sehen Sie da weiteren Handlungsbedarf, was das robuste Design angeht oder funktioniert es?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Das ist ja heute so, dass man mit Big Data ist das Stichwort, dass man die XXXXXXX recht gut überwachen kann und das auch macht. Aber wir haben XXXXX Rückfluss und Informationen an die Entwicklung, XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXX
Das wäre eine ganz wichtige Info. XXXXXXXXXX, bei welcher Anwendung, in welchem Zustand werden Teile geschädigt? Zum Beispiel. XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Zumal heute ja die Möglichkeit besteht.

…

TEIL 2

[53:58] **Was uns jetzt interessiert ist, ob sie für einen der beiden Forschungsansätze oder auch beide ein Anwendungsfall im Unternehmen sehen.**

Ich habe ein Beispiel:
Das habe ich mir von meinem Kollegen geben lassen.

…teilt den Bildschirm...

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Wie kann man sowas verhindern, dass man jetzt sag ich mal bestimmt ein Jahr am Falschen entwickelt? Und sich nicht gleich die Lösung, auch nicht durch die Experten, abzeichnet am Anfang? Sondern eine andere Lösung, die weit weg ist von dem, was man dann am Schluss gemacht hat.

[1:00:12] **Da ist jetzt natürlich als erstes einmal die Frage woran lag das? An welchen Punkten musste man das Konzept dann ändern?**

Also das Problem bei der Lösung war, dass es von der Montageseite und von Fertigungsseite abgelehnt wurde, zwecks der Komplexität. Sie sehen das ja an den Bauteilen - Ich scroll noch einmal hoch - das Bauteil ist ja wesentlich komplizierter. Es ist auch schwerer, war XXXXXXXXX schlechter und natürlich dann auch in der Festigkeit schlechter. Also es hat eigentlich sehr viele Nachteile gehabt. Das einzige, was vielleicht ein Vorteil war, dass wir XXXXXXXXXXXX gehabt haben, um das Ganze abzudecken.

Hier sehen sie dann auch auf der rechten Seite, die XXXXXXXXXXX, die hier sehr groß geworden sind. Und das war von der Fertigung her bei den Zulieferern ein bisschen Aufwand, aber lieber XXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX, wenn man so will, als die andere Lösung. Weil die Lösung, die sie jetzt sehen, das war die, nach allen Kriterien, die wir aufgestellt haben, eigentlich die Bessere. Das wäre jetzt ein Beispiel, das mir einfallen würde, um hier mit ihrem System vielleicht die erste Lösung gleich zu verhindern, wenn das möglich wäre.

Absolut, ja. Das ist ein sehr gutes Beispiel auch, weil man da sieht, dass das Konzept dieses XXXXXXXX jetzt nicht alleine stattfinden kann. Man muss sich frühzeitig die Schnittstellen anschauen, ganz klar und frühzeitig auch Anforderungen bekommen von den verschiedenen Schnittstellen und da ist auch eine Herausforderung oder besser Voraussetzungen, die in den Unternehmen wichtig sind für uns, dass da Anforderungen auch kommen von den verschiedenen Beteiligten.

[1:02:31] **Zum Beispiel, dass die Montage das erste Konzept abgelehnt hat. War Ihnen da die Voraussetzungen von deren Seite her bewusst?**

Ich glaube, das war zu unscharf formuliert. Das wird schon das Problem gewesen sein. Wir haben nicht gewusst, ob das montierbar ist. Das war vielleicht das Hauptproblem. Wir haben das zuerst gebaut, praktisch einen Prototyp gebaut und dann haben wir einen Montageversuch gemacht und haben festgestellt, dass das eigentlich in der Taktzeit nicht montierbar ist.
Das lag wahrscheinlich schon auch daran, dass wir zu wenigen Informationen hatten zu dem Zeitpunkt. Sie haben das ja vorher schonmal angedeutet. "Haben Sie alle Informationen gehabt?" In dem Fall war die (Information) wahrscheinlich mangelhaft.

Also man muss frühzeitig versuchen, so viele Informationen wie möglich zu schaffen. Anders funktioniert unser Ansatz auch nicht. Aber wenn die Informationen da sind und vor allen Dingen, wenn man weiß, wo Informationen gebraucht werden. Und das sieht man dann, wenn man sich die Variablen ansieht und schaut, wie hängen die miteinander zusammenhängen, dann kann man damit auch Variablen lokalisieren, die keinen Zusammenhang haben. Und dann muss man Zusammenhang schaffen.

Das ist ein sehr gutes Beispiel für die Datenbank auf jeden Fall.

Ich hab mir das auch schon gedacht, dass das ein gutes Beispiel wäre, was Sie mal aufsetzen könnten.
...
Ich versuche, mal an die Daten ranzukommen.
...
Ich versuch mal, eine Präsentation zu machen. Dann sehen sie praktisch den Ursprung und das, was man am Schluss dann gemacht haben, dann sehen Sie die Unterschiede und vielleicht ein paar Kommentare dazu. Und wenn es dann noch Fragen gibt, gerne. Dann können wir nochmal drüber reden.

[1:05:17] **Kostenoptimierte Produktfamilien Auslegung war das andere Thema. Sie hatten mal am Anfang angesprochen, das auch geschaut wird, dass diese Komponenten kostengünstig kombiniert werden. Gibt es da auch Handlungsbedarf?**

In zwei Richtungen. Das eine ist das, was wir gerade besprochen haben, mit sehr früh über so ein System mit der Entwicklung drüber gehen.

Das andere war das zweite Beispiel mit Kostenreduzierung über Anzahl der Komponenten, oder? Ist das korrekt?

Ja genau.

Als Beispiel wäre mir das, was ich gerade gesagt habe eigentlich eher eingefallen als die Varianten. Das ist schwieriger greifbar momentan, für mich zumindest.

[1:06:21] **Ich habe noch eine Frage bezüglich Kostenmodell. Gibt es so etwas, worauf sie zugreifen können? Also können die Entwickler auf bestehende Kostenmodelle zugreifen aus früheren Projekten? Oder auch so etwas wie Materialkosten, Herstellkosten... Wie transparent ist das?**

Wir haben eine eigene Abteilung, die Kostenanalysen durchführt, ca. XXX Leute. Die ist In der Konstruktion angesiedelt und die begleitet uns praktisch bezüglich Herstellkosten, Zielkosten, Kostenoptimierung. Margin Improvement Programm haben wir da zum Beispiel. Und das hat in der Vergangenheit sehr gut funktioniert, sodass die Entwicklungskosten schon optimiert wurden. So kann man glaube ich sagen, verschieden stark je nach Baureihe. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Wir haben eine Datenbank, um auf Ihre Frage zurückzukommen. Wir haben Daten, wir haben Kostenabschätzungen. Die sind eigentlich gut zugänglich, die sind auch verfügbar. Somit wissen eigentlich die Entwickler sehr gut, wo wir kostenmäßig liegen und wie die Zielkosten aussehen in dem Projekt.

Danke. Das war wirklich sehr interessant. Also danke auch für das Beispiel.

Wenn man selber dabei ist, dann denkt man nach einem Projekt, das abgeschlossen ist: Was kann man besser machen? Das war auf jeden Fall Verschwendung von Kapazität und Geld. Wenn bei der ersten Lösung soweit daneben liegt. Das sticht dann schon ins Auge und wir haben dann auch die XXXXXXX umbauen müssen mit viel Aufwand und XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX, weil das nicht vergleichbar war, die erste Lösung mit der zweiten.

Ansonsten was ich noch fragen wollte: Was mir noch nicht ganz klar ist. Wir haben hier heute eine bestimmte Vorgehensweise in der Firma, wie wir Lösungen finden. Wie wir jetzt ein Produkt auf den Markt bringen können, wo die Kundenzufriedenheit nach allen Umfragen relativ hoch ist. Indem wir zum Beispiel wie ich am Anfang gesagt habe, unsere Experten in Teams zusammenfassen und dann über Diskussion und weitere Informationen Sammlung oder Bindung dann nach unseren Gesichtspunkten zur besten Lösung kommen. Das kann man natürlich auch so machen, wie sie es jetzt vorher vorgestellt haben, mit... ja ist das ein Expertensystem? Wahrscheinlich kann man das so bezeichnen.

Was ich jetzt nicht weiß, was ist da für ein Aufwand dahinter, wenn Sie jetzt, sagen wir mal so ein Bauteil wie XXXXXXXXXXXXXXXXXXX in so ein System bringen und somit die Lösung auf dem Weg herbringen?

Oder in einem Satz gesagt. Kosten/Nutzen. Wie muss ich Aufwand reinstecken, um vielleicht früher ein besseres Ergebnis zu kriegen?

Also erst mal dazu, was sie gesagt haben, dass es ein "Experten Ansatz" ist. Wir bezeichnen es eher als ein "modellbasierten Ansatz". Es muss kein Expertenwissen vorliegen. Weil Experte würde ich jetzt sagen, ist immer auch mit Erfahrung verbunden, sondern eher eine systematisierte Entwicklung.

Und da hängt es ganz, ganz stark ab, wie die Voraussetzungen im Unternehmen sind und die bisherigen Strukturen. Daher auch einige meiner Fragen vorhin mit der quantitativen Formulierung von Anforderungen zum Beispiel. Wie viele Informationen liegen vor bereits zu Beginn, bevor man das Design startet? Und je nachdem, wieviel Informationen da vorhanden sind, kann es relativ schnell gehen oder relativ lange dauern. Oder zum Teil, wenn Informationen gar nicht verfügbar sind, ist es auch nicht möglich.

Aber von dem, was ich so rausgeholt habe von dem, was Sie gesagt haben, ist es durchaus ein Anwendungsfall, den man sich anschauen kann. Und der Aufwand ist tatsächlich sowohl Branchen als auch Unternehmens abhängig.

Was wir jetzt auch versuchen werden mit den Ergebnissen, die wir aus diesen Interviews bekommen, dann auch einschätzen zu können, wie groß der Aufwand jeweils wäre.

...

Freut mich auch, dass da Interesse besteht von ihrer Seite. Wenn wir wissen wie wir weiter vorgehen, informiere ich sie gerne.

Das wäre gut, ja. Und ich schicke Ihnen die Präsentation mit XXXXXXXXX.

…

AUSSAGEN UND HYPOTHESEN

[1:14:35] **Das sind nochmal Aussagen und Hypothesen, die wir zusammengestellt haben. Da würde ich Sie bitten, dass Sie das jetzt noch kurz bewerten. Sie können gerne Fragen dazu stellen oder sonst einfach aus dem Bauch raus bewerten.**

Der Ansatz der automatisierten Modellierung ist für uns relevant: Das ist genau das, was wir diskutiert haben. Ich glaube, dass das stark Stückzahl-abhängig ist und bei XXXXX vielleicht noch höher gewichtet wird, als bei uns. Ich habe es schon gesagt, wir haben XXXXXX, haben XXXXXXXXXXXX und bei XXX glaub ich werden XXXXXXXXXXXXXXX hergestellt pro Jahr in Deutschland alleine glaube ich. Habe ich mal gehört.
Deshalb gebe ich da mal so einen mittleren Wert. Drei. Wegen der Stückzahl.
Aber vielleicht liege ich da auch falsch.

[1:16:09] **Weil Sie glauben, dass sich der Aufwand dann kostenspezifisch nicht lohnt.**

Aufwand und Nutzen. Man sagt ganz gern, wenn jemand XXXXXXXXXXXXXX entwickelt, der muss auf jeden Cent schauen. Das müssen wir nicht. Weil der Kostendruck nicht so hoch ist und die Stückzahl es auch nicht erlaubt.

Ja ist klar, in der frühen Phase werden die Kosten zu 80% definiert, das weiß man ja und die Entwicklungszeit wird immer kürzer.

Automatisierte Modelldarstellung: hier würde ich wiederum die vier geben, weil man ja wie sie gesehen haben über Umwege mit unseren Teamstrukturen und den Experten natürlich auch zu einer Lösung kommt. Aber es ist natürlich weniger Aufwand, wenn man das mit automatisierter Modellerstellung macht. Das stimmt schon.

Die Produktfamilien ... Herausforderung für die Zukunft

Ich würde die 3 geben, weil das bei unserer Firma, dadurch dass wir stark unterschiedliche XXXXXXXXXXX haben, trifft das bei uns nicht so zu. Sie können nicht bei XXXXXXXXXX, der XXXXXXXX, auf eine Produktfamilie zugreifen von einem XXXXXX, die XXXXXXX. Also die Unterschiede zwischen den XXXXXXXXX sind sehr groß.
Wenn ich das mit der Produktfamilie richtig verstanden habe…

[1:19:15] **Haben Sie schon richtig verstanden. Natürlich stellt sich die Frage, ob man nur denkt, die Unterschiede seien groß aber es dann doch Gemeinsamkeiten gibt, die technisch möglich wären, die man aber nicht kennt. Aber bei Ihnen ist ja auch jedes Produkt individuell.**

Ja, das ist ein bisschen das Problem, dass die Baureihen so stark unterschiedlich sind in unserem Standard und ich spreche auch nur für unsere Standards.
Eben zwischen XXXXXXXXXXX. Wenn Sie jetzt eine XXXXXXXXXXXXXX auslegen, wenn man das mal annimmt und hier die Verwendung über die Brauereien ausdehnen möchte, dann wird es ganz schwierig, weil der Kleine hat eben XXXXXXXXXXXXXXXXXXXX oder ein XXXXXXXXXXXXXX. Also ist die XXXXXX sicher unterschiedlich, wenn man das mal an dem Beispiel festmachen darf.

Super, danke. Nochmal vielen Dank für die Einblicke.

Nr. Interview	08	Unternehmen	X	Datum und Ort	04.06.2020, München	Interviewer	XXX
Zoom Meeting, audio- und visuell (im Homeoffice in XXXX), keine Störungen und angenehme Atmosphäre, sehr aufgeschlossen und redefreudig. Variantenmanagement im Vorhinein ausgeschlossen, da kein Bezug vorhanden. Dauer: 01:01:00							

[01:48] **Für mich zu Beginn, um zu wissen, in welche Richtung wir mit den Fragen gehen, will ich dich bitten, dass du dich kurz vorstellst, das Unternehmen, indem du arbeitest und deine Tätigkeit dort.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[04:17] **Also als Laie gesprochen, ihr entwickelt XXXXXXXXXXXXXXXXXXXXXXXXXXX.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

BEGINN HAUPTTEIL

[04:42] **Hier siehst du unsere fünf Themen, auf die wir uns speziell fokussieren, wo Herausforderungen in der Produktentwicklung unserer Meinung nach entstehen. Ich habe das Varianten Management jetzt von vornherein ausgeklammert, weil du ja schon geschrieben hast, dass ihr gar nichts damit zu tun habt. Also, es gibt keine Varianten bei euch?**

Also, es gibt natürlich immer Varianten, soweit gehe ich jetzt schon mal. Natürlich gibt es verschiedene Entwürfe, also jedes Feature bekommt ein Software Design und geht durch die Produktarchitektur und dann gibt es natürlich vielleicht mal zwei unterschiedliche Entwürfe. Aber das, was jetzt der klassische PEler unter Variantenmanagement versteht mit Varianten Bäumen aufzubauen und so weiter. Also jetzt wie XXXXXXXXX zusammen stöpseln, das ist natürlich überhaupt nicht für uns der Fall. Deswegen habe ich es ausgeklammert, weil wir da ein ganz anderes Verständnis davon haben.

[05:35] **Ja, ich glaube, das ist dann auch gut, dass wir da lieber die Zeit für andere Fragen verwenden. Wenn du dir die anderen Begriffe anschaust: Anforderungsmanagement, Komplexität, robustes Design und Entscheidungsfindung sind das Themen, die euch auch betreffen?**

Anforderungsmanagement natürlich ständig jeden Tag. Ich rede den gesamten Tag mit Kunden, weil wir natürlich Anforderungen erfüllen müssen. Ansonsten tut es die Konkurrenz.

Komplex sind unserer Produkte definitiv, wir leben in einer komplett vernetzten Welt. Unsere Produkte laufen nicht allein, sondern im Verbund mit vielen anderen.

Entscheidungsfindung: Wenn 80 Leute, die im Team entwickeln, in die falsche Richtung laufen und etwas entwickeln, was halt eben am Schluss am Markt vorbeigeht oder nicht die Hauptanforderung erfüllt, ist es natürlich ziemlicher Mist. Das sollte nicht passieren, sonst sind wir bald aus dem Geschäft.

Und Robustes Design ist die Grundanforderungen. XXXXXXXXXXXXXXXXXXXXXXXXXXXX. Wir haben einen XXXXXXXXXXXXXXXXXXXX von 100 Prozent. Das heißt, XXXXXXX garantiert seinen Kunden bei dem Delivery

Part unseres Businesses, dass wir nicht ausfallen. Nie, niemals. Und das seit die Firma existiert, ist es auch noch nicht passiert. Das, wir haben Boards, die bei jeder Entscheidung mitreden, die sich nur darum kümmern, robust zu sein und zwar absolut hundertprozentig robust. Also wie ein Flugzeug. Wir dürfen nicht runterfallen.

[07:14] **Fallen dir weitere Herausforderungen ein, die da jetzt nicht genannt sind?**

Dynamik, extreme Dynamik!
Meine Produkte sind ständig unter Feuer. Das heißt, wenn wir heute eine neue XXXXXXXX Technologie rausbringen, was wir jede Woche machen, dann dauert es manchmal 15 Minuten, bis die ersten Bösewichte XXXX anfangen gegen dieses Ding zu schießen. Und das ändert sich ständig. Wir haben eine Horde von weltweit vernetzten Leuten, die aktiv gegen uns arbeiten, so schnell es geht. Deswegen ist das ein ständiger Kampf, ein Katz-und Mausspiel, so läuft das dauernd. Und deswegen ist alles, was wir vor einem Jahr gemacht, heute mehr oder weniger obsolet, weil es sowieso schon längst bekannt und umgangen wird. Wir müssen also unglaublich hohe Dynamik an den Tag legen. Entscheidung werden täglich revidiert.

[08:25] **Und auch Änderungen dann dementsprechend XXXXXXXXXX, an dem Produkt?**

Natürlich genau. Also gerade gestern saß ich bis Mitternacht mit einem Team zusammen, weil wir eine XXXXX hatten, die gestern Abend reinkam. Also ich habe im Schnitt XXXXXXX von Kunden, die eben nicht nur als normalen Service bearbeitet werden können, sondern dann eben von Spezialisten gemacht werden müssen - pro Woche.

…

[09:44] **Ich würde dich bitten, dass du die Begriffe, die du siehst, in eine Reihenfolge bringst der Relevanz nach. Dann gehen wir demnach noch mal die Themen einzeln durch.**

1. Anforderungsmanagement 2. Dynamik 3. Robustes Design 4. Komplexität 5. Entscheidungsfindung.

ANFORDERUNGSMANAGEMENT

[10:56] **Ganz allgemein, welche Herausforderungen treten bei der Herstellung von Anforderungen auf?**

Kunden: Die Anforderungen kommen logischerweise vom Markt, die muss man in irgendeiner Weise gewichten, weil es natürlich immer mehr Anforderungen gibt als die, die man tatsächlich umsetzen kann. Und Kunden sind da nur bedingt hilfreich. Entweder sie bringen mir nicht die Anforderungen, die sie tatsächlich brauchen. Also, da ist sehr viel Interpretation dabei. Entweder ich kriege falsche Informationen oder natürlich überhaupt keine Priorisierung, weil alles ist wichtig.
Dann auch ganz häufig hört man Anforderungen, die hinterher, wenn sie tatsächlich eingeführt werden, plötzlich überhaupt keine Rolle mehr spielen. Nicht mehr relevant sind. Die Frage ist man braucht ein unglaublich guten Kundenzugang, um zu bewerten, von wem diese Anforderung eigentlich kommt. Also ob das jetzt ein Entscheidungsträger ist, der etwas möchte, der aber am Schluss nur Folien sieht oder ob das jemand ist, der im täglichen Geschäft sitzt als XXXXXXXXX, der dieses Ding für die tägliche Arbeit braucht.
Und dann die Anforderungen gegeneinander zu gewichten. Also ich habe XXXXX Kunden, die alle verschiedene Anforderungen haben. Dann kann ich natürlich auch einfach anfangen zu zählen. Jetzt bringt der eine Kunde natürlich XXXXX mehr als der andere Kunde, wie schaut es dann aus. Kunden haben dann auch teilweise jahrelange Beziehungen zur eigenen, also zu unserer Firma und XXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX. Der ändert dann plötzlich Prioritäten. Ja, weil er halt nun mal diesen Menschen kennt.
Das heißt der ganze Prozess, wenn man erstmal die Anforderungen sinnvoll zusammenbringt, weil man am Schluss die Sachen ja mehr oder weniger sequenziell abarbeitet. Das hat mit dem Management noch gar nichts zu tun, sondern erstmal mit der Sammlung. Ich kann nur an einer begrenzten Anzahl von Projekten gleichzeitig arbeiten, ohne dann hinterher das robuste Design zu gefährden. Das ist ausgesprochen schwierig.

Das war jetzt erstmal die Sammlung, dann spielt das Ganze mit dem zusätzlichen Punkt der Dynamik zusammen, dass sich jeden Tag Prioritäten verschieben und zwar einfach aufgrund bei uns XXXXX. Schönstes Beispiel:
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Wie geht man mit solchen Sachen um? Also die Frage, kann ich das relativ schnell umsetzen? Wieviel Design brauche ich dafür? Kann ich das Unabhängig von allen anderen Sachen machen? Diese Dynamik, die durch ständig auftretende Erscheinungen im normalen Geschäft stattfindet, die ist ausgesprochen schwierig.

[14:42] **Wie verwaltet ihr die Anforderungen?**

Ach, da gibt es unterschiedliche Ansätze. Tatsächlich auch von Produkt zu Produkt unterschiedlich. Ich persönlich nutze weitestgehend Jira. Einfach weil wir Engineering-lastig sind und wir sind ursprünglich eine reine Engineering Firma.
(XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Also Chefs und Chefinnen sind natürlich gemeint. Immer wenn ich hier eine männliche Form verwende, sind natürlich beide Geschlechter gemeint.)
Mein XXXXXXXXXXXXXX kommt halt ursprünglich aus dem reinen Engineering. Das heißt, die kennen natürlich solche Tools wie Gira, viel, viel besser, als wenn ich jetzt irgendwas anderes hernehmen. Letztendlich ein Tool ist ein Tool. Es gibt auch welche, die nehmen dann AHA her, weil das ja so schön für Produktmanager geeignet ist. Dann ist es natürlich klasse für mich als Produktmanager. Aber ich muss das Zeug ja auch kommunizieren können zu anderen Leuten. Und so eklig Jira teilweise ist in der Handhabung, so bekannt ist es halt eben auch unter den Entwicklern. Es geht halt darum, dass die Leute auch draufschauen. Deswegen, so ein Backlog ist eigentlich das Normale zum Sammeln der Ideen und dann wird einmal im Quartal alles mal rausgekramt, jedenfalls die Langfristziele, und priorisiert.

[16:38] **Werden die Anforderungen eher qualitativ oder quantitativ erfasst?**

Guter Punkt...
Ich weiß, woher das quantitativ kommt, nämlich aus der klassischen Konstruktion. Ja, wie dick muss das Blech sein und wie groß.

[17:03] **Kann man da Vergleiche ziehen?**

Unsere Anforderungen werden schon ausgesprochen genau beschrieben. Und wenn ich sage, wir beschreiben das mit Jira, dann stimmt das nicht ganz. Es ist natürlich immer die Kombination Jira und Confluence, also die (...)Suite halt komplett. Das heißt am Anfang kommt eine relativ grobe Use-Case Beschreibung, das ist das allererste. Allerdings wird, bevor wir das Ding überhaupt priorisieren können, festgelegt, ob dafür ein Produkt Architekt gebraucht wird. Und Produkt Architekten bringen schon so viel Input. Also das ist immer noch eine normale Textbeschreibungen, aber in der Textbeschreibung sind einfach schon so viele Details enthalten, dass basieren darauf ein Ingenieur, ein Softwareingenieur das Ding komplett in Code umsetzen kann.

Also von daher würde ich sagen, die Dinger sind zu hundert Prozent quantitativ.
Weil hinterher der Coder eigentlich weiß, was er zu tun hat.

[18:13] **Und nicht mehr viel Spielraum hat.**

Richtig, eigentlich keinen Spielraum. Das heißt, es geht so weit, dass man aus diesem Produkt Architekten Design die Zeit, die benötigt wird, bestimmt wird. Also der Umsetzungsaufwand, der ist daraus eigentlich sehr, sehr genau zu bestimmen und da reden wir halt von auf die Woche genau teilweise bis auf den Tag genau.

ROBUSTES DESIGN

[18:46] **Ich würde jetzt die Dynamik mal überspringen, weil da hast du eingangs schon was gesagt dazu. Was mich noch interessieren würde, ist jetzt mehr das Thema Robustes Design und Komplexität. Fangen wir mit dem robusten Design an.**

Wie werden denn eure Produkte bewertet? Also was sind Bewertungskriterium für robustes Design?

Da wird es halt extrem technisch. Also wir haben hier keinen Katalog. Das ist nicht wie in der Mechanik, wo es klare Designregeln gibt, die in einem Katalog zusammengefasst sind. Natürlich gibt es Software Designregeln und wie Softwarecode geschrieben werden muss. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX. Aber wir haben verschiedene Entscheidungsboards, es gibt ein Architektur Board, das entscheidet, ob die Architektur denn robust ist, ob die Architekturen stimmt und auch in der Zukunft keine Probleme machen wird. Wir haben ein XXXXX Board, was Informationssicherheit bewertet. Wir gehen durch mehrere Freigabe Prozesse, in denen alle beteiligten Produkte sitzen. Allerdings gilt das nur für ein Teil unserer Designs. Also, das heißt XXXXXXX gehen da nicht durch, einfach weil sie viel zu dynamisch sind. Aber wenn ich ein Projekt mache, was mindestens sechs Monate dauert, dann geht das durch einen Prozess, indem ich ein Konzept Commit brauche, indem ich ein XXX Commit brauche, indem ich ein XXXXXXXXX genehmigt haben muss. Und da redet eben jedes Produkts plus eben die Boards. Das heißt Architecture Design muss abgesegnet sein. XXXXX muss abgesegnet sein, und dann wird es freigegeben. Also ein relativ aufwendiger Prozess.

[20:53] **Also, du meintest ja auch, dass am Ende die Produkte hundertprozent robust sind. Und das ja auch eine Anforderung ist. Aber auf dem Weg dahin siehst du da Verbesserungsbedarf oder Handlungsbedarf oder hast du da zu wenig Einblick dafür?**

Hoffentlich habe ich genügend Einblick, ich bin ja der Produktmanager. Es wäre schlimm, wenn ich im Blindflug unterwegs wäre.
Klar gibt es da Verbesserungsbedarf. Absolut. Und zwar aufgrund der Dynamik.
Ich meine, das ist die Historie unserer Firma. Wir kommen aus einem Feld, was absolute Robustheit benötigt, aber eben auch relativ viel Veränderung sieht. Ich meine, jemand, der sich heute ein XXX kauft bei uns, der schließt den Vertrag ab und dann sehen wir uns im Idealfall in drei Jahren wieder, wenn's um die Verlängerung geht. Und dazwischen werden Bits und Bytes durch die Gegend gestorben. Das sollte möglichst ohne Störung passieren.

Jetzt ist die XXXXXXX Welt halt nun mal so, dass sich jeden Tag etwas ändert. Ich habe, seit ich diesen Job mache, fast keine Woche erlebt, in der nicht was aufgetreten ist, was wir noch nie vorher gesehen haben. XXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Wir machen klassisch keinerlei Änderungen während dieser Zeit. Das heißt da ist Moratorium, da ist kein Software Release. Klar, all jene Software, die sich ändert, kann irgendwie kaputt gehen. XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXDas heißt, die Verheiratung von robustem Design mit hoher Dynamik ist theoretisch oder klassisch gesehen unvereinbar. Und da steckt das größte Entwicklungspotenzial drin meiner Meinung nach. Wie die neuen Prozesse laufen sollten.

KOMPLEXITÄT

[23:59] **Du hast vorher schon ein bisschen gelacht, als du Komplexität gesehen hast.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXWas einfach heißt, dass kompliziert/komplex wird einfach bei uns als Worthülse benutzt.
Daneben auf der Folie steht an Anforderungsmanagement. Und da haben wir eigentlich eine relativ klare Definition und ein sehr, sehr einheitliches Verständnis. Das fehlt uns halt bei Komplexität total. Das heißt, wenn wir hier über Herausforderungen "Komplexität" reden, dann ist das eine große Blase.

[25:31] **Okay, also ist vielleicht jetzt tatsächlich auch hier auf der Folie einfach nur so hingeklatscht. Worauf sich die Fragen beziehen ist die Produkt Komplexität, also die Strukturen innerhalb des Produktes. Wie wirken sich gewisse Änderungen auf andere Komponenten, auf andere Variablen aus. Das Gesamtsystem Produkt, ob das verstanden ist, im Unternehmen oder ob gerade zwischen Abteilungen die Schnittstelle klar sind.**
Tatsächlich eine genaue Definition kann ich da jetzt auch gerade gar nicht geben, worauf wir uns beziehen.

Deswegen ist es ja auch Komplex. (lacht)

[26:11] **Wie ist es denn im Softwarebereich? Gibt es da auch Strukturen des Produkts, der Software am Ende, die von allen verstanden werden und wie Änderungen stattfinden? Du hattest vorher genannt 80 Entwickler arbeiten da dran...**

Nur an diesem einen Produkt. Ich weiß ehrlich gesagt gar nicht, wie viele Produkte wir anbieten. Aber es ist mindestens XXXXXXXXXXX Zahl an Produkten. Und die hängen alle zusammen. Und jeder Kunde kann sich jedes Produkt kaufen und oder fast ausnahmslos jedes Produkt kaufen und kombinieren. Und das Ganze setzt auf, auf einem XXXXXXXXX Produkt. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Allein die Datenbank, die wir haben.
Also ein XXXXXX Algorithmus für XXXXXX hat einerseits natürlich die Formulierung, wie der Algorithmus funktioniert und die Variablen darin werden ja befüllt. Und diese Variablen hängen davon ab, was im Markt draußen ist. Also Beispiel die Kombination von validen Geräten. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXX Ja dann stimmt da was nicht. Jetzt haben wir zick solcher Merkmale. Welche Kombinationen sind valide und welche sind nicht valide? Das ändert sich bei uns minütlich, weil wir das gesamte Internet durchforsten. Das heißt Sachen wie, da kommt ein XXXXXX, XXXXXXXXXXXXXXXXXXXXXXXXXXXX, von XXXX entwickelt. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Und das muss natürlich in unseren Algorithmen, in den Variablen berücksichtigt werden. Das passiert natürlich ständig und laufen. Das ändert aber natürlich das Verhalten unserer Software und zwar im Minutentakt. Also wir bringen hier ja nicht irgendwie eine Karre raus, die wird dann nach vier Jahren gefaceliftet und nach acht Jahren wird sie dann ersetzt. Wir machen solche Änderungen laufend, die gesamte Zeit.

[29:28] **Da schwebt die Dynamik über allem ja auch wieder.**

Richtig. Und es macht eben nicht nur mein Produkt, sondern das machen andere Produkte auch.
Und klar, es gibt kleine Änderungen, es gibt große Änderungen, die gemacht werden. Aber letztendlich im Softwarebereich gerade bei solchen Themen, wird ständig released. CICD ist ja eins von diesen Schlagwörtern, also kontinuierliche Entwicklung, kontinuierliche Integration. Und das passiert halt im Mechanik Bereich nicht, denn was ich einmal rausgebracht habe, das hängt halt beim Kunden an der Wand. Und das ändert sich nicht mehr.

[30:11] **Oder muss auch über Jahre abgeschrieben werden und man will auf gar keinen Fall, dass ich was ändern muss.**

Genau. Und das ist halt bei uns überhaupt nicht der Fall.
Allein unsere XXXXXXXX basiert auf einer grundlegenden Konfiguration für den Kunden. Und je nachdem, welches XXXXXX Produkt er dazu kauft, nutzt das dieselbe Plattform. Das heißt, die Produkte laufen zusammen auf derselben XXXXX Basis. So und da greift jedes Produkt darauf zu. Diese Basis ändert sich ja auch. Natürlich gibt es Modul Grenzen, aber es muss viel, viel dynamischer gehandhabt werden als im mechanischen Bereich, wo ich halt nun mal meinen Modularen Querbaukasten habe und der besteht jetzt erst mal eine Zeit lang.

…

TEIL 2

[41:33] **Eigentlich stelle ich an dieser Stelle die Frage, ob es in der jeweiligen Firma Anwendungsbedarf für einen der beiden Ansätze in der jeweiligen Firma gibt. Ich nehme an, das ist bei euch jetzt zu abstrakt, dass ihr unsere Ansätze anwenden könntet.**

Ich meine, Wirknetze, wir stellen uns ja alle vor, dass unsere Hirne auch so arbeiten. Es gibt natürlich ein paar Masterbrains, auch bei uns in der Firma, die zumindest Großteile des Zusammenspiels unserer verschiedenen Softwarekomponenten verstehen. Ja, die mag es mit Sicherheit geben. Und die braucht man auch. Ich kenne einen und ich kenne seine Telefonnummer auswendig, weil für die harten Fälle braucht man solche Leute. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXX. Und offensichtlich komme ich ja auch damit ganz gut klar, das heißt mit Schnittstellen und so weiter klappt es offensichtlich ganz gut bei uns.

Dass man dann Sachen da drin, dass man ganze Produkte retired und austauscht durch andere, so wie bei euch dann eben ganze Komponenten, ist natürlich auch möglich und wird auch gemacht. Ich meine, die Frage ist nur wie komme ich zu diesem Wirknetz und das geht halt meiner Meinung nach im mechanischen Bereich deutlich leichter als im Softwarebereich. Und zwar aus dem simplen Grund, dass wenn ich mechanisch etwas ändere, dann hängt da immer ein Rattenschwanz dran. Da habe ich Kosten, das Zeug muss ich herstellen. In der Software kann ich's halt relativ schnell austauschen. Und deswegen spielt es keine so große Rolle. Und ist eben auch viel schwerer zu definieren.
Wenn wir eine konzeptionelle Änderung in einem Produkt vornehmen, dann ändert sich natürlich irgendwas an dem Wirknetz. Aber ich kann das theoretisch nächsten Monat wieder ändern. Und es klappt halt, weil es eben auch nicht so extrem teuer ist. Klar, ich brauche halt ein paar Entwickler und ein Produkt Architekten, aber ich habe im Wesentlichen nur die Mann-Kosten und nicht die ganze Mechanik Geschichte hinten dran.

Vorhin habe ich so ein Auto gesehen. Wenn XXXX halt nun mal irgendein Krosserie Blechteile am Schluss baut, und es baut es halt, weil es abhängig ist von allen anderen Komponenten, also eine Motorhaube. Ja dann müssten die halt auch das Werkzeug bestellen für das Presswerk. Und da gehen halt einfach dann die riesigen Millionenbeträge hin. Das ändere ich halt hinterher nicht mehr so ohne weiteres. Besser gesagt, ich ändere das gar nicht mehr. Und deswegen sind diese Sachen so unglaublich wichtig im Mechanik Bereich, aber eben halt auch leichter eingrenzbar. Also ich kann halt die ganzen Anforderungen und Restriktionen viel besser erfassen als im Softwarebereich. Das ist jedenfalls meine Meinung.
Es gibt, die Wirknetze jetzt bei uns ganz genauso. Aber zum einen sind sie halt flexibler und zum anderen sind sie halt - Na ja, warum ein Produkt Architekt eine gewisse Entscheidung getroffen hat, das ist halt, weil er dachte, dass das das Beste ist. - Also es ist wahrscheinlich schwieriger zu erfassen.
Schön wär's, sowas zu haben. Ich stelle mir echt nur die Frage, wie komme ich in unserem Bereich dazu, so ein Netz überhaupt erstmal zu haben? Und erst dann habe ich ja den großen Benefit davon.

[45:16] **Ja ok, aber wenn das Netz aufgestellt wäre, könnte man ja auch Änderungen, die laufend reinkommen, dort wieder einspeisen und sieht die Änderungen auf andere Bereiche.**

Absolut. Und wie gesagt, es gibt wahrscheinlich diese paar Leute bei uns im Unternehmen, die tatsächlich große Teile dieses Wirknetzes im Kopf haben. Wie kriege ich das Ganze jetzt in ein System rein?

Gestern hatte ich so ein Fall. Ein Kunde ist der Meinung, er will XXXXXXXXXXXXX mit meinem Produkt absichern.

[45:55] **Okay, was sind XXXXXXXX?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Ja, das steckt halt in den Köpfen von ein paar Leuten drin, wie kriege ich es raus?

Also vielleicht, das Einzige, was ich anmerken wollte ist, es wäre schön zu wissen, wie die Erstellung dieser Netzwerke optimal gestaltet wird. Denn das ist der Aufwand, der getrieben werden muss, damit ihr den Nutzen rauskriegt, der definitiv gegeben ist, wenn ich es mal habe.

[47:27] **Was siehst du denn für weiteren Nutzen?**

Der Nutzen ist halt eben ganz klar. Dass ich daraus QA Pläne erstellen kann und dass ich Änderungsausbreitungen sehe. Also es heißt, was passiert, wenn ich an einer Stelle etwas ändere? Was ist die Auswirkung davon? Dass ich also Risiko abschätzen kann. Änderungsaufwände und Risiko. Und dass ich die Sachen eben sinnvoll absichern kann. Weil das ist doch das, was meistens passiert, ich ändere an einer Stelle und habe eben nicht alle Konsequenzen im Blick und hinterher knallt's halt irgendwo.
Das kann ich im Softwarebereich relativ simpel abfangen. Klar, es knallt halt bei irgendeinem Kunden aber mit einem XXXXX, den ich halt mache, ist sowas relativ schnell repariert. Im Automobilbau habe ich dann einen Rückruf, der kostet mehr.

[48:26] **Also man kann agiler auf das Feedback vom Markt reagieren. Und das will man ja vielleicht auch erst mal, oder? Wenn man so eine Software released, werde nicht auch Kosten gespart, indem man erst mal schaut, wie es läuft. Und dann sich dann den Rücklauf anschauen und anpasst? Oder ist das kein schönes Vorgehen?**

Naja, im Idealfall natürlich nicht.
Praktisch, natürlich gibt es Beta Phasen. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Und dann wird halt noch einmal nachgebessert. Aber man sollte natürlich schon… Das wichtigste ist die Architektur in Entscheidungen. Wenn ich die am Anfang falsch gewählt habe, dann kann ich mein gesamtes Design wegschmeißen und dann verlieren auch wir viel Geld. Von daher sollte das sich jetzt nicht komplett ändern.

[49:27] **Du hast gesagt, das steckt in den Köpfen der Leute. Also es gibt an sich jetzt keine vergleichbare Datenbank für den Softwarebereich?**

Nicht, dass ich wüsste.

[49:57] **Du hast als Schwäche noch genannt, dass das Wirknetz natürlich erstmal aufgebaut werden muss. Also erstmal lokalisiert werden muss, wie sieht das Wirknetz aus. Siehst du andere Schwächen oder Verbesserungspotenziale des Ansatzes?**

Also ich versuche jetzt natürlich, die Schwachstellen herauszufinden. Also nimm es mir nicht übel und ich will hier nichts schlecht reden.
Ich bin da vollkommen resistent.
(lacht) Ok. Also ich meine diese Idee dieses automatisierten Designs ist ja jetzt schon alt. Ich meine, diesen Wunschtraum, den gibt es ja schon lange. Die Frage ist warum bist du denn bis heute so nicht realisiert worden? Oder warum nur in manchen Bereichen?
An was ich gerade gedacht habe ist - mir fällt nur der Name gerade nicht ein. - Ich hatte mal zu tun mit einem Entwicklungsleiter bei XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Und die haben sowas schon in den frühen Zeiten des CADs, da haben die schon damit angefangen, solche Verletzungsgeschichten aufzubauen und haben in diese ganze Auslegungs- und Planungsgeschichte bereits die Produktion selber mit einbezogen. Also, das heißt alle Restriktionen, die die hatten, die haben wirklich im CAD, wenn die entwickelt haben, dann bereits constraints aufgezeigt, weil gewisse Sachen eben dann auf eine andere Produktionsmaschine springen müssten, dann werden sie teurer. Das ist so dass, was am nächsten an das rankommt, was ich hier sehe. Und das war wirklich toll, weil deren Gewinn lag darin, dass sie für jede Anwendung XXXXXXXXX so optimal designen, dass sie in der Produktion gewinnen, XXXXXXXXXXXXXXXXXXXXXXXXXX
Ich weiß nicht, wo die heute sind mit sowas. Aber vielleicht wäre das auch noch ein ganz hervorragender Spielpartner für euch, gerade weil eben auch XXXXXXXXX mechanisch relativ schön zu beschreiben ist. Die Anforderungspunkte sind sehr schön zu beschreiben. Ich meine, ich kann wunderbar unterscheiden, XXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX. Das ist mir gerade noch so in den Sinn gekommen.

Vielleicht nicht als Schwachpunkte, aber eben als Teil, der berücksichtigt werden muss. Es wird immer weniger rein mechanisch. Und wenn ihr die Software mit einbinden wollt:
Die Software schafft mir ja Flexibilität. Ich meine, das sieht man ja, dass dieselben Motoren in drei verschiedenen Ausbaustufen verbaut werden und gesteuert wird das Ganze über Software. Software gibt mir halt relativ viel Freiräume und ich weiß nicht, inwieweit ihr diese Spreizung von Grenzbereichen oder von constraints abbilden könnt in den Wirknetzen. Weiß ich einfach nicht, habe ich keinen Einblick.
Habe ich auch zu wenig Einblick, muss ich ehrlich sagen.
Also das wäre halt noch eine interessante Sache.
Im Wirknetz heißt halt einfach, wenn ich an dieser Schraube drehe, dann passiert halt in der Auswirkung auf irgendein anderes Element dies und das. Und wenn ich eine andere Blechdicke nehme, dann habe ich halt eine andere Steifigkeit. Nachvollziehbar. Aber so ein Software Baustein schafft mir halt Flexibilität. Mit demselben Softwarebausteine kann ich halt den gleichen Motor bei meiner Bohrmaschine für völlig verschiedene Anwendungen hernehmen.
Dann könnte ich im Prinzip einen haben.
Ja, im Extremfall. Das ist ja das, was bei Motoren passiert. Die werden künstlich beschnitten. Ich produziere denselben Motor, aber ich habe ihn halt eben in drei verschiedenen Ausbaustufen dann vorhanden. Inwieweit das berücksichtigt ist, vielleicht ist das drin. Ich meine, es ist naheliegend. Wahrscheinlich werden sich da schon schlauere Leute als ich Gedanken gemacht haben. Das wäre vielleicht insgesamt noch zu sehen.

[54:47] **Ja, ich denke auch darüber nach, dass das dann ja aber auch wieder eine Kostenfrage ist. Ist es günstiger, ein Motor herzustellen und den mit Software zu regeln? Oder nimmt man doch drei unterschiedliche, die dann nicht geregelt werden müssen?**

Genau...

AUSSAGEN UND HYPOTHESEN

[55:26] **Hier haben wir noch mal ein paar Aussagen gesammelt. Da würde ich dich bitten, dass du die mal kurz einordnest.**

Produktfamilienauslegung ist für mein Unternehmen relevant: Nee. Ich meine, wir haben vom selben XXXXXX, dann vielleicht zwei oder vielleicht sogar mal drei verschiedene Varianten, je nachdem, ob das halt mal ein kleineres oder größeres Unternehmen ist. Aber das war es dann auch schon.

Der Ansatz der automatisierten Modellierung ist für mein Unternehmen relevant: Nee, nee. Das wird alles sehr individuell gebaut.

Wie relevant bewerten sie den Fokus auf die frühe Phase: Der Ausdruck der frühen Phase… das ist witzig. XXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX "Man muss die frühe Phase als erstes betrachten." Es ist ganz wichtig. Ich meine und die Antwort ist immer gleich: Natürlich. Ich meine, wenn ich die frühe Phase nicht beachtet, dann läuft halt hinterher alles andere auch schief. Ist ja klar. Also: Ja! Die frühe Phase war vor 20 Jahren wichtig. Das ist sie heute auch noch.

Der Fokus des Solutions Space Ingenieuren Ansatzes auf die frühe Phase ist richtig gesetzt:
Wenn ich einen Solutions Space betrachten will, dann muss ich ihn in der frühen Phase betrachten. Woher stammt diese Kurve mit den Änderungskosten vs. Freiräumen, die ich habe. Ich meine, die wurde mir als Student auch schon gezeigt. Also folglich muss es ja wohl eine Fünf sein.

Automatisierte Modelldarstellung mit modularen Produktmodellen verringert Modellierungsaufwand:
Absolut.
Und wie gesagt, ... Ich schaue mal, ob ich den Namen noch finde. XXXXXXXXXX und ich hatte da einen guten Kontakt. Ich habe mit dem man zusammengearbeitet. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Die Produktfamilien Auslegung ist eine entscheidende Herausforderung für die Zukunft:
Nein, macht da jetzt wenig Sinn. Es kommt halt aufs Produkt an. Ich mein bei unseren Produkten nicht. Dann bleiben wir doch mal neutral. Ich meine, wie sagt es kommt halt drauf an wo. Es kommt echt darauf an wo. Der XXXXXXXXXXXX braucht es bestimmt. Bei mir spielt es keine Rolle.

In der Zukunft werden quantitative Analysen die Produktentwicklung maßgeblich steuern:
Hm..,Ich nehme da jetzt auch mal eine Drei. Einfach, weil mir die Frage zu ungenau ist. Weil halt eben Produktentwicklung. Wenn ihr da jetzt hingeschrieben hätte: "Die mechanische Produktentwicklungen", da hätte ich völlig zugestimmt.
Also ich meine, ich sitze hier in XXXXXX in einer Welt, die jede Woche XXXXX rausbringt und es ist halt eben ganz, ganz wenig, was noch diese reine Mechanik ist. Hier sind die ganzen XXXXXXXXXX zum Beispiel. Die haben natürlich unglaublich viel Mechanik. Aber wenn ich mir anschaue, was die für Leute suchen, dann ist da unglaublich viel Programmierung dabei. Ich kenne zwei Programmierer, die in den XXXXXXXXXX arbeiten und das sind riesige Zentren. Da spielt die quantitative Analyse wahrscheinlich weniger eine Rolle. Also die Frage ist, wie viel der Produktentwicklung ist denn noch die mechanische Produktentwicklung. In der Mechanischen, da ist das, was ihr sagt, wahrscheinlich alles absolut zutreffend.

Super, danke für das Interview.

Gerne, wenn ich irgendwo noch weiterhelfen kann, dann gerne.

Nr. Interview	11	Unternehmen	X	Datum und Ort	09.06.2020, München	Interviewer	XXX
Angenehmes Gespräch über Zoom, nur audio. Entspannte Gesprächsatmosphäre mit genug Zeit nach hinten Video nur halb gesehen. Wenig Vorkenntnisse über den Ansatz, aber selber Modell-basierte Planungsmodelle XXXXXXXXXX betreut/entwickelt. Daher gutes allgemeines Verständnis. Robustes Design weggelassen, da nicht relevant in Bezug auf Herausforderungen. Dauer: 01:16:11							

[02:24] **Ich weiß bisher, dass du bei XXXXXXXXX arbeitest und die Stichworte, die mir noch gegeben wurden, waren XXXXXXXXXXXXXXXXXXXXX. Stell dich bitte kurz vor und was du genau machst.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[05:00] **Ok, also ihr vergleicht auch die Konzepte der verschiedenen Hersteller?**

Vergleichen ja. Aber das ist dann eher in der XXXXXXXXXXXXXX. Ich bin in der Entwicklungsphase tätig, da ist dann wirklich nur noch ein Hersteller. Natürlich kann man es vergleichen mit irgendwelchen Vorgängerprojekte oder ähnlichen, aber es geht eher um das Bewerten. Also ist es so, wie wir uns das vorstellen. Gibt es irgendwelche Probleme? Wir möchten auch die Erfahrung der Ingenieure oder vom Betrieb mit einfließen lassen, ist die Lösung so zielführend? Oder gibt es da, wenn es so umgesetzt wird, immer wieder Probleme? XXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[06:09] **Ok, die Freigabe ist dann das Ende sozusagen?**

Das ist dann das Ende der Entwicklung, beziehungsweise bis zur Aufnahme. Dann geht es noch in die Gewährleistungsbetreuung. Da geht's eigentlich nur darum die Sachen zu klären, die auftauchen. XXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX. ..

BEGINN HAUPTTEIL

[07:11] **Du siehst hier auf der Folie fünf verschiedene Stichworte/Herausforderungen, die wir in der frühen Phase angesiedelt sehen. Ich sag kurz was dazu.**
...
[08:17] **Fallen dir spontan noch weitere Themen ein, die darunter nicht fallen?**

Vielleicht erst mal noch eine allgemeine Information oder Vorbemerkung.
Es ist nicht so das klassische, wie jetzt XXXX zum Beispiel, dass wir entwickeln und dann ist es da und wir verkaufen es. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Das heißt, wir haben gewisse Berührpunkte mit wahrscheinlich allen von den Punkten. Aber halt nicht so im klassischen Sinn. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Anforderungsmanagement - auf jeden Fall.
Entscheidungsfindungen in gewissem Maße.
Varianten Management vielleicht weniger, aber auch im gewissen Maße dann XXXXXX.
Komplexität auf jeden Fall.

Robustes Design sind bei uns dann eher Sachen, die wir zwar betrachten und in die Anforderungen mit reinbringen. Aber das ist etwas, dass der Hersteller umsetzen muss.
Bei uns ist das alles nicht so das Klassische. Deswegen also im Hinterkopf für dich, dass bei und nicht so der klassische Fokus ist, wie bei einem normalen herstellenden Betrieb.

[09:54] **Du darfst gerne, wenn du auf Fragen nicht antworten kannst oder das Gefühl hast, so wie es die Frage stelle, kannst du sie nicht beantworten, dann hacke da gerne jederzeit ein.**

Also Herausforderungen, sind halt bei uns insbesondere solche Designaspekte wie: Sicherheit, Zuverlässigkeit und Qualität. Steckt wahrscheinlich irgendwo bei eurem robusten Design drin.

Und was bei uns auch eine ganz große Herausforderung ist, was uns jetzt auch von vielen anderen Bereichen unterscheidet, ist die extrem lange Lebenszeit. Also, dass der Lebenszyklus XXXXXXXXX im Gegensatz zum Beispiel zum Auto schon mal von vornherein auf mindestens XX Jahre ausgelegt ist. Wenn ein Auto XX Jahre alt ist, ist es ein Oldtimer und XXXX, XXX kann teilweise nach XX Jahre noch mal ein facelift / redesign bekommen und dann geht's wieder weiter. Dementsprechend ist das auf jeden Fall etwas, das bei uns eine große Herausforderung ist und wo man halt dann wirklich auch in die Zukunft schauen muss.

[11:21] **Was sind das für Herausforderungen, die dadurch entstehen, dass ihr so lange Lebenszyklen habt?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Also solche Sachen hat man halt bei der Auslegung von so einem Gerät einfach nicht betrachten können. Deswegen ist die lange Lebenslaufzeit wirklich eine große Herausforderung bei uns.

[12:24] **Diese ganzen Herausforderungen. - Wenn ich das jetzt richtig verstanden habe - dann bestehen sie darin, wie formulierte ich meine Anforderungen an die Hersteller, oder? Weil ihr selber macht technisch XXXXX nichts? Auch nicht nachrüsten?**

Also nachrüsten, ja.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...

[14:06] **Klicke mal bitte auf den geschickten Link. ... Sortiere die Begriffe (Herausforderungen) bitte mal der Relevanz nach. So, wie sie bei euch am wichtigsten im Unternehmen sind oder am häufigsten vorkommen. Robustes Design haben wir jetzt schon ausgeschlossen. Du kannst es gerne auch mit ranken, also wie es für dich wichtig ist. Aber da reden wir dann nicht drüber.**

1. Anforderungsmanagement, 2. Entscheidungsfindung, 3. Komplexität, 4. Variantenmanagement, (5.Robustes Design)

ANFORDERUNGSMANAGEMENT

[15:00] **Dann fangen wir mit dem Anforderungsmanagement an. Die Fragen beziehen sich darauf, wie Anforderungen strukturiert werden. Da ist erstmal die Frage, gibt ihr nur Input als Anforderungen oder strukturiert ihr auch, seid ihr Teil der Vorentscheidungen was sind genaue Systemanforderungen und so weiter?**

Das Ergebnis der ersten Phase bei uns im Beschaffungsprozess, sind wirklich die fertigen Bedingungsunterlagen mit allen Anhängen, mit dem Lastenheft. Dementsprechend geht es wirklich drum, wir formulieren das Ganze, wir schreiben das Lastenheft, wir klären das mit entsprechenden Personen ab.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX nur, dass man das jetzt einfach hinschreibt, sondern was bedeutet das für die Kosten, was bedeutend das für die Wartung und so weiter.
Und schreiben das dann letztendlich auch ins Lastenheft rein. Also, das ist ja nicht nur eine Systemanforderungen auf das XXXXXXXXXXXXXX, sondern das beziehen sich dann auf Untersysteme und so weiter.

Was wir auch noch machen ist, wir haben eine eigene Abteilung in meinem Bereich. Die heißt Anforderungs- und Ausschreibungsmanagement und die beschäftigen sich dann auch mit der Strukturierung solcher Lastenhefte und sogar mit sowas wie in Richtung Automatisierung in Anführungszeichen. Also, dass die Lastenhefterstellung beschleunigt werden kann und alles was da an Stichworten rumläuft, Strukturierung, Modularisierung, Wiederverwendung und so weiter, das ist da alles in dem Bereich.

[17:33] **Okay. Und teilte ihr auch in die verschiedenen Ebenen ein? Also habt ihr zum Beispiel Basisanforderungen und dann Systemanforderungen, Komponentenanforderungen...**

Ja, was verstehst du unter Basisanforderungen? Und was ist der Unterschied zu Systemanforderungen?

Also für mich sind Basisanforderungen sowas wie XXXXXXXXXXXXXXXXXXXXXXXXXXX können. Und dann wäre eine Systemanforderungen speziell zum Beispiel XXXXXXXXXX, dass die nur so und so viel XXXXXXXXX dürfen. Das wäre dann für mich eine Systemanforderungen für XXXXXXX. Also dass du die Anforderungen so formulierst, dass einzelne Komponenten spezielle Anforderungen bekommen, die aber verknüpft sind mit der Basisanforderung XXXXXXXXX zum Beispiel.

Ok. Also wir unterscheiden da nicht zwischen den beiden Bereichen, also wir haben ein Systemlastenheft, das quasi beides umfasst. Und dann haben wir spezielle in Anführungszeichen Unterlastenhefte, je nachdem wie groß das Projektes ist. Und da hast du das dann wirklich auf Module Ebene. Und für so ein Modul, da gibt es dann auch eine Norm, nach der das alles unterteilt und beschrieben ist. Und du kannst dann halt sagen, für jeden Bereich - bei uns heißt das dann Gewerk. - für jedes Gewerk hast du dann ein eigenes Lastenheft. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Das ist dann glaube ich auch das, was du als Komponenten Ebene bezeichnet hast.

[19:42] **Genau. Okay, und innerhalb dieser Anforderungen, werden die hauptsächlich qualitativ formuliert. Oder gibt es auch viele quantitative Anforderungen? Also quantitativ im Sinne von wirklich konkret Zahlenwerten.**

Da uns die Anforderungs-Formulierungsregelungen auch geläufig sind, werden sie natürlich bewertbar gestaltet. Insbesondere, weil ja mein Job dann ist, zu bewerten, ob die Anforderungen erfüllt sind. XXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Und das kannst du nicht bewerten. Dementsprechend müssen wir es natürlich qualitativ ausgestalten.

[20:34] **Und das funktioniert auch?**

Ja. Wir haben eine eigene Abteilung, die da nochmal den Finger darauf hält und die sind da im positiven Sinne sehr pedantisch und die legen großen Wert drauf. Und alle unsere Fachingenieure, die dann letztendlich für die einzelnen Gewerke die Anforderungen schreiben, sind da auch getrimmt drauf. Und ja, es geht sogar so weit, dass da schon in Richtung Anforderungsschablonen gearbeitet wird und Formulierungsregeln und so weiter. Das wird auf jeden Fall gehandhabt. Was natürlich nicht heißt, dass in der Realität nicht doch das ein oder andere mal so eine Anforderungen dann irgendwie durchrutscht. Aber unterm Strich wird es schon so gelebt, und eben nicht so die klassischen quantitativen Ansätze verfolgt.

[21:31] **Kannst du mir eine Prozentzahl sagen, wie viele Anforderungen quantitativ erfasst sind in den Lastenheften?**

Theoretisch keine. Praktisch, also in Prozent ist das schwer auszudrücken. Einzelne, wie gesagt, jeder, der die Anforderung schreibt kann dann im Umkehrschluss wieder dafür verantwortlich gemacht werden oder wird verantwortlich gemacht, dass er es prüft. Dementsprechend liegt es in seinem eigenen Interesse, dass er es bewertbar formuliert.
Wir haben es eh schon gesagt, dass wir bestimmte Bestrebungen in Richtung Modularisierung und ähnlichem verfolgen. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[22:34] **Und wie funktioniert die Kommunikation mit XXXXXX? Gibt es da häufig Zielkonflikte?**

(Ziel-) Konflikte, ja.

Tatsächlich haben wir ja schon einmal grundsätzlich einen unterschiedlichen Fokus. Wir wollen natürlich am besten alles haben und da wir ja im Wettbewerb sind, es läuft ja über XXXXXXXXXXXXXXXXX, dass wir auch möglichst günstig sein müssen. Wir wollen zwar die besten Leistungen haben, aber halt auch für möglichst wenig Geld und der Hersteller will natürlich möglichst wenig von seiner Plattform abrücken. Bei uns gibt es den Begriff: Wir kaufen von der Stange. Was es aber in der Realität nie ist. Du kaufst nie XXXXXX, der eins zu eins genauso ist, wie das, was der Hersteller anbietet. Aber der Hersteller versucht eben in die Richtung zu gehen und wir wollen, insbesondere weil unsere Kunden, beispielsweise XXXXXXXXXXXXXXXXXXXX, immer irgendwelche Sonderwünsche haben. Und das ist dann von Region zu Region unterschiedlich, XXXXXXXXXXXXXXXXXXX. Ich betreue jetzt zum Beispiel bei XXX fünf verschiedene Projekte, und es ist immer der gleiche XXXXX. Aber es beginnt damit, dass er dann in XXXXXXXXXXXX anders ausschaut. Dass er anders lackiert ist und ein anderes Innendesign hat, dass die einen irgendein System wollen, die anderen nicht. Die nächsten wollen dann irgendeine Software Update und so weiter. Es ist nie das gleiche und das ist schon mal grundsätzlich der Zielkonflikt.

Wo auch nochmal Zielkonflike auftreten ist, wenn der eine den Vertrag so auslegt und der andere den Vertrag so. Dann geht es darum, muss das jetzt umgesetzt werden oder nicht? Und wenn ja, wie kann man das integrieren und so weiter?

Und wo es dann auch immer wieder Probleme gibt, wenn man während der Entwicklung neue Wünsche hat. Das heißt, alles was mit Änderungsmanagement zu tun hat. Wir stellen irgendeinen Änderungsantrag und dann geht es auch darum, passt das rein. Was muss man alles betrachten und so weiter. Und da kann es dann auch sein, dass der Hersteller sagt, es geht nicht so, weil es ist sicherheitsrelevant und so weiter. Das müssen wir so lösen, dann ist es wieder zulassungsrelevant und so weiter. Das ist immer, wo man dann zwischen den verschiedenen Fronten unterwegs ist.

[25:31] **Okay, also ihr seid ja sozusagen Kunden bei den Herstellern. Aber ihr habt wiederum andere Kunden, wo ihr auch wieder Wünsche berücksichtigen müsst.**

Ja, und das ist tatsächlich... Wenn ein Projekt startet, dann gibt es immer so schöne Kommunikations-Charts. Wer mit wem und überhaupt. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

ENTSCHEIDUNGSFINDUNG

[27:09] **Gut, wir gehen mal weiter in die Entscheidungsfindung. Was für Entscheidungen betrifft das denn bei dir, wo Herausforderungen entstehen. Und was sind das für Herausforderungen?**

Fangen wir mit dem Großen an. Es beginnt damit, wir müssen herausfinden, welcher Hersteller für uns der Beste ist, XXXXXXXXXXXXXXXXXXXXXXXX

Die Entscheidung wird natürlich dadurch verkompliziert, dass wir an XXXXXXXXXXXXXXXXX gebunden sind XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXX Und dann kann sie jeder darauf bewerben und derjenige, der das beste Angebot macht, der wird es dann. Da geht dann die erste Entscheidung los, welche Hersteller wird es. Da haben wir dann auch als Unterstützung so eine Entscheidungsmatrix, wo dann die einzelnen Anforderungen gewichtet sind und so weiter. Dass man das nachvollziehbar gestalten kann und warum ist jetzt der genommen worden und nicht ein anderer? Das sind so die ersten Entscheidungen und das sind auch dementsprechend die größten.

Dann gibt es Entscheidungen hinsichtlich welcher Systeme oder welche Funktionen werden umgesetzt oder nicht? Beispielsweise welche Optionen werden gezogen. Wenn wir das Lastenheft aufschreiben, dann gibt es sogenannte Muss-Optionen, Soll-Optionen, Soll-Anforderungen und so weiter. Das heißt, die haben unterschiedliche Umsetzungswege. Eine Muss-Option ist zum Beispiel eine, wo man ein Angebot verlangt vom Hersteller und wenn das Angebot passt, dann wird die Option gezogen. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Bis hin dann zum Kleinen. Also wie ich schon gesagt habe, wenn irgendwelche Design Prüfungen anstehen, müssen wir wiederum sagen, ob es das ist, was wir verlangt haben. Ist die Lösung richtig? Brauchen wir vielleicht noch irgendwie zusätzliche Sachen? Oder können wir das freigegeben? Das ist dann vom ganz Großen bis hin zum Kleinen diese Entscheidungen, die zu treffen sind.

[30:57] **Wie ist bei den kleinen Entscheidungen das Vorgehen?**

Wir haben als Entscheidungsunterstützung die Informationen, die uns der Hersteller gibt. Sei es in PDF Form bis hin zu irgendwelchen Berichten. Oder was ich auch schonmal gemacht habe XXXXXXXXXXXXXXXXXX, dass man wirklich XXXXXXXX hingeht und schaut, ist die Anforderung erfüllt. Das Vorgehen ist entsprechend, du hast die Aufforderung, die wie auch immer gestaltet ist. Schaust im einfachsten Fall auf ein Dokument, wo irgendein Prüfbericht oder sonst irgendwas abgebildet ist. Da steht es ist erfüllt. Dann ist die Entscheidung einfach. Dann kannst du sagen, ja passt, da kannst du verweisen drauf, fertig. Im schwierigeren Fall ist es dann so, du musst etwas interpretieren oder vielleicht XXXXXXXXXXX sehen.
Aber jetzt direkt einen Entscheidungsprozess an sich, wüsste ich jetzt nicht, dass wir ein Vorgehen haben. Aber es ist halt so das klassische Vorgehen. Du kriegst irgendwelche Informationen zur Verfügung gestellt, auf Basis derer du dann halt die Entscheidungen treffen musst.

[32:10] **Gibt's da Informationen, die häufig fehlen und die hilfreich wären?**

Kann ich jetzt so direkt nicht sagen. Also, dass da immer welche fehlen, nein. Weil in dem Beschaffungsprozess, bestimmen wir ja welche Informationen wir verlangen. Das ist dann Vertragsgegenstand. Das heißt, die müssen uns das liefern. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX dann kann es durchaus sein, dass die Unterlagen, die man bekommt, nicht in der entsprechenden Qualität sind. Damit ich versuche irgendwie die Frage zu beantworten. Das ist dann das, woran es eher scheitert. Aber nicht, dass jetzt irgendwelche Unterlagen einfach nicht da sind und wir deswegen die Frage nicht beantworten können.

...

KOMPLEXITÄT

[33:50] **Jetzt sprechen wir noch kurz über Komplexität. Habt ihr denn damit zu tun, dass zum Beispiel die Auswirkungen auf das Gesamtsystem nicht ganz klar sind, wenn ich einzelne Komponenten ändere oder wenn ihr - weil du vorher schon angesprochen hast, du hast in XXXXXXXXXXXXXXXXXXXX verschiedene Anforderungen jeweils wieder auf XXXXXX. - ist da transparent, wie sich gewisse Änderungen dann auf das Gesamte auswirken?**
Oder ist für dich Komplexität was anderes?

Also normalerweise so Änderungsauswirkungen, die sind eigentlich meistens klar. Insbesondere weil wir, wie vorher schon gesagt, für jedes Gewerk Experten haben, die XXXXXXXX besser kennen als ihre eigene Westentasche, im Extremfall. Es kommen auch immer wieder welche aus XXXXXXXX zu uns, dementsprechend kennen die dann teilweise auch XXXXX das Produkt an sich sehr gut.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Die Komplexität ist bei uns eher anders zu verstehen. Also Komplexität an sich ist schon mal das System XXXXXXXXXXXX per se.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Ich habe es nicht an erster Stelle priorisiert, weil es nichts ist, was ich direkt verändern kann, sondern ich muss auf die Umgebungsbedingungen mit dem Projekt reagieren. Aber es ist jetzt nicht irgendetwas, dass ich jetzt speziell Möglichkeiten zur Komplexitätsreduktion anwenden würde.

[38:52] **Auch ein interessanter Einblick. Das würde ich vom Thema her auch eher ins Anforderungsmanagement mit reinbringen.**
…

VARIANTENMANAGEMENT

[39:11] **Variantenmanagement, nur kurz. Betrachtet ihr Varianten also verschiedene XXXXXXXXX? Wie ihr da mit den Herstellern in der Zukunft standardisieren könnt? Oder Produktfamilie schafft ...**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Wir haben halt eine XXXXXXX, sie aus verschiedensten XXXXX besteht und irgendwie sollten die auch wieder zusammenarbeiten können. Dementsprechend gibt es jetzt da irgendwie zu betrachten, wie hängt das zusammen und welche Varianten kaufen wir ein?
Es wäre natürlich am schönsten und am einfachsten für den Betrieb, für die Wartung und Instandhaltung, für die Werkstatt und so weiter. Wenn wir einfach immer nur XXXXXXXXXXX hätten. Wo man dann einfach nur gewisse Verbesserungen haben, in der Theorie passt alles zusammen, alles wunderschön und wunderbar.

[41:28] **So stellt es sich der Laie auch vor.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX macht der ein besseres Angebot und so weiter. Und dementsprechend haben wir von x-verschiedenen Herstellern, x-verschiedene Modelle und die sind untereinander nicht kompatibel. Und so weiter.
Und dann wird es interessant.

...

[42:34] **Die Präsentation, die ich dir jetzt zeige, ist eher aus Sicht des Herstellers. Du kannst aber vielleicht im Hinterkopf behalten, auch später für die Bewertung, wie es wäre, wenn einer eurer Hersteller das Verfahren verwenden würde. Ob das euch auch etwas bringen könnte. Beziehungsweiße natürlich auch, ob XXXXXXX sich mit so einem Tool anfreunden könnte.**

...

TEIL 2

[1:00:03] **Fangen wir erstmal von der Seite her an: Siehst du ein Anwendungsgebiet bei euch selber im Unternehmen?**

Also als ich es das erste Mal gehört habe, habe ich mir gedacht: nein. Also als du den Überblick das erste Mal vorgestellt hast.
Jetzt würde ich sagen, also es spiegelt ja schon das, was ich gesagt habe, wider, dass gewisse Größen da sind und dann muss man halt abwägen, welche Variante man dann tatsächlich auswählt. Also jein.

[1:00:53] **Würde es denn helfen, wenn die Hersteller sowas verwenden?**

Ich überlege jetzt gerade, wer tatsächlich in welcher Situation welchen der Ansätze anwenden könnte.

Also kostenoptimierte Auslegung sehe ich jetzt nicht bei uns. Hersteller natürlich. Ich mein, da geht es ja genau in die Richtung. Da haben wir ja das schöne Schlagwort: große externe Varianz bei geringer Interner. Und das heißt, du willst mit möglichst wenig Komponenten irgendwie möglichst viel aufdecken oder mit Plattformstrategien und Sonstigem. Also beim Hersteller sehe ich das auf jeden Fall, dass die das kostenoptimiert auslegen. Insbesondere, weil halt wahrscheinlich jeder irgendwie irgendwelche Sonderwünsche hat und dementsprechend das mit möglichst wenig verschiedenen Komponenten oder Varianten abzudecken, das ist auf jeden Fall ein Ziel, das ich beim Hersteller sehe.

Die automatisierte Modellerstellung der Zusammenhänge. Das sehe ich in gewissem Maß bei uns, aber bei uns halt nur auf einer geringeren Ebene. Der Hersteller hat da halt nochmal viel mehr, was er betrachten muss. Bei unseren Lastenheften steht jetzt nicht drin bis zur letzten Schraube, wie komplett das ausschauen soll, sondern wir haben da Angaben im Sinne von, das System soll das und das können, die Komponente soll das und das können.

[1:02:40] **Das Abstraktionslevel von den Datenbanken oder von den Modellen selber, das kann man ja selber wählen. Was deine Stellgrößen sind, bestimmst du selbst.**

Genau. Aber wir haben natürlich viel weniger, als der Hersteller und sind, wie du sagst, auf einem andern Abstraktionsniveau, also wir unterhalten uns irgendwie über XXXXXXXXXXXXXXXXXXXXXXXXX und der Hersteller geht da halt noch viel tiefer rein.
Dementsprechend anfänglich könnte ich mir es tatsächlich vorstellen, dass wir uns halt solche Sachen anschauen. Aber es ist die Frage und da sind wir jetzt wieder bei der Killerfrage für alles, XXXXXXXXXXXXXX XXXXXXXXXXXXXX wie sehe ich, ab wann es sich lohnt? Ich habe da einen gewissen Zeitaufwand. und jetzt sind wir wieder bei der klassischen Aussage: "Ja unsere Experten wissen des ja a so." Und damit kannst du ja jeden Ansatz, XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX sofort platt machen.

Also für den Hersteller lohnt es sich wahrscheinlich schon, ob das jetzt bei uns - um irgendeine Zahl zu nennen - mit den zehn Varianten oder Variablen, die wir uns da anschauen, ob sich das da lohnt jetzt ein Modell aufzustellen... Insbesondere müssten wir ja dann die ganzen Zusammenhänge kennen. Weil, wie ich das jetzt verstanden habe, um das mit Matlab oder mit ähnlichem irgendwie rechnen zu können, muss ja da überall eine Formel dahinterstecken und dementsprechend brauche ich da die Zusammenhänge zwischen den einzelnen. Und jetzt sind wir wieder beim Einmalaufwand, der sich dann auszahlt.
Das heißt, ich muss da mindestens einmal viel Zeit reinstecken, um die ganzen Zusammenhänge in Formeln zu bringen und das Modell zu erstellen und so weiter. Und natürlich kannst du dann wie immer erklären, ja beim zweiten, beim dritten und beim 25. Mal durchlaufen lohnt sich das Ganze. Ich habe aber jetzt in der Realität

festgestellt, es ist schon schwer, für ganz einfache Tools die Zeit zu finden, um die zu implementieren und dann auch zu verwenden.
Das heißt zu argumentieren, dass wir so etwas anwenden wird wahnsinnig schwer, zumindest in meinem Bereich. Was aber nicht heißt, dass ich deswegen den Mehrwert nicht sehen würde oder sehen könnte.

[1:05:14] **Siehst du denn mehr das Problem der Bereitschaft der einzelnen Beteiligten, sich mit dem Tool auseinander zu setzen oder liegt es im Hinblick auf das Ressourcenmanagement, die fehlende Zeit sich damit auseinander zu setzen?**

Ein bisschen was von beidem. So eine kleine Anekdote am Rande. Wir führen gerade Teams ein. Und wie schwer es schon ist, ein vermeintlich einfaches Tool, das man für ganz grundlegende Sachen anwenden kann, sich damit zu beschäftigen. Und mit dem Tool arbeitest du ja jeden Tag eigentlich, die ganze Zeit und wie schwer es da schon ist, dass sich die Menschen nur ein bisschen damit beschäftigen. Und dann übertrage ich das jetzt mal auf so ein Tool.

Dementsprechend auf der einen Seite die Bereitschaft und auf der anderen Seite halt auch die Ressourcen. Genau das, was ich gesagt habe. Natürlich lohnt es sich bei einem fünften oder zehnten Durchlauf. Aber wir haben ja unsere Stunden Projekt spezifisch und da ist es wahnsinnig schwer, da jetzt zu sagen, wie finanziere ich mich jetzt, wenn ich das Tool anwende.
Das müsste man dann irgendwo schon ins Angebot reinschreiben, dann müsste ich die Stunden und so weiter und so fort...
Das heißt es ist eine Kombination aus beidem, dass man die Ressourcen nicht haben. Meistens finanzieller Natur, was dann in der Zeit resultiert und auch die Bereitschaft. Es gibt zwar Bestrebungen, gewisse Sachen einfach einzuführen und das zu versuchen, aber das ist halt nochmal auf einem ganz anderen Niveau. Also da geht es jetzt ganz stark auch nochmal um die Selbstorganisation, also um irgendwelche Stundenberechnungen und ähnliches. Das heißt, da sind wir noch ein ganzes Stück davon entfernt, sowas dann tatsächlich anzuwenden.

[1:07:35] **Gut, dann sind wir schon fast am Ende. Ich danke dir schon mal für deine interessanten Einblicke.**

...

AUSSAGEN UND HYPOTHESEN

[1:08:28] **Bewerte die (Aussagen und Hypothesen) bitte mal so, wie du sie siehst, jetzt vor oder nach dem Gespräch, egal, wie es sich gerade anfühlt, so aus dem Bauch raus. Kannst aber auch gerne was dazu sagen.**
Es ist dann für uns nur nochmal eine Tendenz am Ende.

Ok. (hat abgeschickt)

Relevanz für das Unternehmen ist auf jeden Fall da. Aber wie gesagt, mit dem Hintergrund, dass die Umsetzung schwierig ist.

Fokus auf die frühe Phase. Also ich meine, ich habe mich ja selber mit der frühen Phase beschäftigt, das heißt, da hat man die größten Entscheidungsmöglichkeiten. Wenn das dann mal in der Entwicklung ist, dann ist das Kind schon sozusagen in den Brunnen gefallen. Und dementsprechend, dass man sowas in der frühen Phase macht, definitiv richtig.

Dass die automatisierte Modelldarstellung den Modernisierungsaufwand verringert, kann ich jetzt momentan einfach nicht sagen. Da kann ich nicht abschätzen, wie viel da dahintersteckt. Ich weiß zum Beispiel nicht, wie kompliziert es ist, dass ich da irgendeinen Zusammenhang in eine Formel bring.

[1:10:59] **Ja, da sind wir selber auch noch am Experimentieren und auch auf der Suche immer wieder nach Anwendungsfeldern.**

Ja. Dementsprechend kann ich es dann noch weniger sagen.

Dass es eine Herausforderung für die Zukunft sein kann, ja, auch das sehe ich schon. Ob das dann tatsächlich in der Zukunft darüber maßgeblich gesteuert wird ist wieder schwierig zu sagen. Mit dem Zusammenhang von Aufwand und Nutzen, was ja bei uns bei jedem Ansatz XXXXXXXXXXXXXXXX wahrscheinlich immer die gleiche Frage ist. Es ist, so wie du mir das jetzt präsentiert hast, würde ich sagen, ja, auf jeden Fall.

Wenn wir dann aber in die Realität schauen, wenn wir jetzt den viel zitierten Elfenbeinturm der Forschung verlassen, dann ist die Frage, wie wahrscheinlich ist es, dass das in der Industrie tatsächlich umgesetzt wird?

Und da sind halt die Hürden, die ich jetzt aufgezählt habe und das denke ich mal ist nicht nur bei uns, sondern das ist wahrscheinlich überall so.
Und dementsprechend kann ich es nicht sagen, also ich weiß von dem ein oder anderen Ansatz XXXX XXXXXXXXXXXXXXXXXX, der dann tatsächlich in der Industrie genutzt wird. Oder zumindest als Beratungswerkzeug verwendet wird, XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Aber es ist halt immer das Problem, wenn man das auf Papier und in einem kleinen Anwendungsfall macht, das wird immer gut aussehen, bei so einer Schlauchtrommel, die kann ich mir theoretischen im Kopf auch überlegen. Wenn ich dann aber wirklich ein kompliziertes System habe, wie XXXXXXXXXXX oder was auch immer, da wird es dann schon wieder viel schwieriger. Und dann kommen wir auch wieder in den Bereich von der Rechenleistung.

Wenn ich mich damit lange genug beschäftige, dann finde ich tatsächlich für jede Variabel, die ich habe, irgendeine Formel, wie die mit den anderen zusammenhängt. Aber dann muss ich das ganze Zeug auch irgendwann mal ausrechnen und dann kann ich mir auch vorstellen, dass das durchaus mit so einem komplizierten Produkt wie XXXXXXXX gar nicht mehr so einfach wird.

Respektive - das ist mir vorher auch irgendwann noch eingefallen. - Da haben wir ja das schöne Schlagwort: Shit in, shit out. Das heißt, wenn ich da irgendwie die Formel nicht anständig gestalte, dann ist eigentlich mein Ergebnis Schwachsinn.

Und das sehe ich da irgendwo als Gefahr oder als Hürde, dass ich das wirklich irgendwann mal anwenden kann. Das sind wir wieder beim Aufwand, weil da muss ich wirklich extrem viel Aufwand in die Modellerstellung stecken und wenn ich das nicht mache, dann brauche ich es gleich gar nicht machen. Und da ich jetzt aufgrund von der Stunde Vorstellung noch nicht beurteilen, wie schwierig da jetzt die Anwendung ist und wie wahrscheinlich es ist, dass das sich irgendwo durchsetzt.
Nichtsdestotrotz sehe ich das als ganze interessanten Ansatz und sehe auf jeden Fall Möglichkeiten, wo man das anwenden kann.

[1:14:30] **Okay. Das war ein gutes Feedback und nochmal eine gute Zusammenfassung.**

...

[1:15:30] **Wenn die Interviews durch sind, überlegen wir mal, einen Workshop zu organisieren, wo wir das dann weiter anwenden, und vielleicht auch konkretere Beispiele anschauen. Darf ich dich da informieren, wenn wir uns in die Richtung bewegen?**

Ja.

Gut, dann bekommst du da nochmal Rückmeldung.

Nr. Interview	15	Unternehmen	X	Datum und Ort	17.06.2020, München	Interviewer	XXX
Zoom, allerdings schlechte Verbindung, daher beidseitig ohne Bild. Keine Zeit nach hinten raus. Dauer: 00:53:12							

[01:36] **Stell dich zu Beginn bitte kurz vor und beschreibe deine Tätigkeit im Unternehmen.**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[03:27] **Und ihr seid auch Teil der Herstellung XXXXXXXXX?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[04:00] **Wie lange bist du schon im Unternehmen tätig?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[04:08] **Und auf derselben Position?**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX ...

BEGINN HAUPTTEIL

[05:27] **Wenn du die Herausforderungen hörst, fällt dir da was ein, was damit gar nicht abgedeckt wird? Wo immer wieder Herausforderungen bei der Entwicklung entstehen?**

Was teilweise nur drin ist, ist, gerade beim Thema Anforderungen, zu verstehen, was die Anforderungen eigentlich sind. Also das ist für mich weniger ein Thema der Formulierung der Anforderungen im Sinne, da gibt es ja diese Bausteine mit XXX und so weiter. Das ist für mich nicht das Thema. Das eigentliche Thema ist für mich, die Anforderungen korrekt zu sammeln und vollständig zu sammeln.

06:08 **Noch weitere?**

Das wär's. ...

[06:30] **Hier sind die fünf Herausforderungen nochmal gelistet. Bring die bitte der Relevanz nach in eine Reihenfolge. Und dann gehen wir die dementsprechend nacheinander durch.**

1. Anforderungsmanagement, 2. Entscheidungsfindung, 3. Robustes Design, 4. Komplexität, 5. Variantenmanagement

---

ANFORDERUNGSMANAGEMENT

Anforderungsmanagement habe ich jetzt so interpretiert, dass das, was ich gesagt habe, da mit drin ist.

[07:31] **Ja ist auch mit drin. Da können wir auch direkt gleich einsteigen.
Gehe da bitte noch einmal darauf ein, was du vorher gesagt hast und erläutere, wo du da die Herausforderung siehst.**

Wie gesagt, die Herausforderung ist für mich bei dem Anforderungsmanagement, die Anforderungen vollständig und korrekt zu erheben.

[07:57] **Und wieso funktioniert das manchmal nicht? Oder was fehlt an Informationen?**

Das Wissen ist halt oft implizit und verteilt. Es kommen oft Anforderungen wie: es soll genauso sein wie die alte Maschine und folgendes zusätzlich. Stellt sich die Frage, was ist denn genauso wie die alte Maschine?

Was sind die genauen Anforderungen? Und dann sind diese Aussagen oft vereinfachend, weil man oft an viele Details nicht denkt. Muss es eine XXXXXX Option geben? Muss es eine Option geben für den US Markt, wo andere Regularien herrschen? Ist es so, dass es heute vielleicht fünf Varianten gibt, aber eigentlich nur zwei benutzt werden und drei gar nicht gebraucht werden? Wenn jetzt die Aussage kommt, genauso wie das alte Produkt, werden wieder alle fünf gebaut. Aber derjenige, der sagt genauso wie das alte Produkt, der hat vielleicht eine andere Vorstellungswelt. Der sieht nicht diese fünf Varianten, die es gibt, sondern er denkt vielleicht nur an die eine, die er immer verkauft.

Weil Menschen nicht unbedingt das gleiche meinen, wenn sie das Gleiche sagen.
Und dann sieht jeder auch nur diesen Bias dahingehend, dass jeder das für am Wichtigsten hält, was ihn gerade bewegt hat in letzter Zeit.

[09:34] **Wie geht ihr dann damit um, wenn ihr so schwammige Aussagen bekommt?**

Gut, man versucht es natürlich genauer zu haben. Fragt nach, man fragt andere. Und letztendlich hängt es, da man die Personen ja auch kennt, mit denen man redet, weil wir das in der Regeln nicht direkt vom Kunden erheben, sondern über den Vertrieb erheben oder auch intern in der Technik erheben, weiß man natürlich, wer da vertrauenswürdiger ist, wer mehr Erfahrung hat. Da ist viel Bauchgefühl dann auch dabei.

[10:14] **Wie werden denn die Anforderungen dann an die Entwickler übergeben? In welchem Stadium befindet sich das - ich nenne es jetzt mal Lastenheft?**

Also in der Regel ist es so, dass das mündlich oder schriftlich in sehr unvollständiger Form übergeben wird, und wir uns dann das Lastenheft erst einmal selber zusammenbauen.
Da gibt es dann auch eine Vorlage mit verschiedenen Bereichen, also auch Entsorgung, welche Normen sind notwendig? Was muss für die Wartung beachtet werden?
Also über so verschiedene Themen, die gerne vergessen werden. Und dann vollständigen wir diese Liste dazu. Wobei auch da die selbstverständlichen Sachen nicht drinstehen, sondern nur die Sachen, die für wichtig empfunden werden.
Das heißt 95 Prozent der tatsächlichen Anforderungen, die es gibt, - Ich habe jetzt irgendeine Zahl gesagt, vielleicht sind es auch 80 Prozent. - stehen da gar nicht drin, weil sie selbstverständlich sind. Und das funktioniert mit erfahrenen Entwicklern sehr gut oder relativ gut, birgt aber immer das Risiko, dass doch irgendetwas vergessen wurde, was eigentlich jedem bekannt ist, aber es eben implizit ist.
Weil es in unserer Industrie nicht so ist wie in der XXXXXXXXXXXXXXX, dass wir 100 Seiten Anforderungsliste haben oder Lastenheft, was notwendig ist, um auch die trivialen Sachen, sag ich mal, festzuhalten, weil sie vielleicht nicht für alle trivial sind, sondern es werden eigentlich nur die wirklich neuen oder entscheidenden Punkte genannt. Was dazu führt, dass man immer wieder Zwischenschritte machen muss und Designreviews oder wie auch immer, um dann mit den anderen Stakeholdern zu klären, ob nicht doch irgendetwas vergessen wurde. Und da kommt dann in der Regel immer wieder was dazu... "Ah ja die Option brauchen wir auch noch, haben wir nicht gedacht das brauchen wir auch.

[12:26] **Also werden Anforderungen bis zum Ende der Produktentwicklung noch variiert oder angepasst?**

Idealerweise nicht. Also ich sag mal in der ganz späten Phase nicht mehr. - kommt zwar auch immer wieder vor - aber ich sage mal gerade in der Entwicklungsphase an sich werden schon noch Anforderungen geändert ja. Oder auch in der frühen Phase kommt es zu heftigsten Anforderungsänderung. Das kommt auch vor. Weil man feststellt, dass diese Anforderung so nicht erfüllbar sind in ihrer Gesamtheit. Es wird dann zu teuer. Und dann werden Anforderungen geändert. Dann startet mal wieder von vorne, was aber zum gewissen Teil auch naturbedingt in der Entwicklung ist, dass man das manchmal vielleicht macht.

[13:18] **Kannst du eine Prozentzahl sagen, wie viel am Ende der Anforderungen quantitativ erfasst ist.**

Bezogen auf die erfassten Anforderungen?

Genau.

Das versuchen wir schon zu treiben. Also ich würde sagen 70 Prozent, eher 80 Prozent.
…

ENSCHEIDUNGSFINDUNG

[14:02] **Bei dem Thema interessiert mich auch ganz allgemein erst mal, welche Herausforderungen während Designentscheidungen auftreten und wie ihr damit umgeht, wie Entscheidungen getroffen werden.**

Wie wir damit umgehen, wie Entscheidungen getroffen werden? Oder wie wir Entscheidungen treffen?

Wie ihr Entscheidungen trefft und dann damit umgeht, falls es Probleme gibt.

Jetzt auf rein technischer Ebene treffen wir die selbst in der Entwicklungsabteilung, sofern die Vorgaben relativ klar sind, was notwendig ist.

Und dann gibt's immer wieder Reviews mit unseren Stakeholdern im Sinne von Auftraggebern für die Entwicklung. Wo wird den aktuellen Stand diskutieren und es dann wie so Meilensteine entweder weiter freigegeben wird, oder man sagt ne, muss nochmal überarbeitet werden.

[15:08] **Wer ist da beteiligt bei diesem Entscheidungsprozess? Also ist immer ein Entwickler verantwortlich für seine Komponente oder gibt es Expertenteams?**

Ja, ein Entwickler ist verantwortlich für seine Komponente. Entscheidung wird getroffen, also im Entscheidungsgremium ist dann der Entwickler der, der vorstellt. Da bin ich dann mit drin. Ich kenne das aber auch schon vorher. Wir besprechen das vorher schon innerhalb der Entwicklung. Und dann noch der Auftraggeber, das ist der Geschäftsführer und technische Leiter, das ist eine Person. Gegebenenfalls noch jemand aus dem Vertrieb, also aus dem Verkauf. Also der echte "Empfänger" dieser Entwicklung.

[15:57] **Ok, du hast Entscheidungsfindung als zweite Herausforderungen genannt. Warum? Was fällt dir noch dazu ein?**

Weil gerade in den frühen Phasen sind es immer Entscheidungsfindung unter hoher Unsicherheit. Ich meine es müssen Entscheidungen getroffen werden, welches Konzept man verfolgt, es ist aber noch nicht klar, ob es funktioniert.
Es ist auch noch nicht klar, wie viel Aufwand zwingend dahintersteckt, weil ja oft der Teufel im Detail sitzt. Das heißt, es müssen oft Entscheidungen unter Unsicherheit getroffen werden.

[16:34] **Habt ihr bei den Entscheidungen schon Berechnungen oder Ergebnisse aus Simulationen zur Verfügung oder auf Basis welcher Informationen wird es dann entschieden?**

Teilweise. Wenn es gerade mechanische Bauteile sind, sind die natürlich oftmals simuliert als Entscheidungsgrundlage oder als Hilfsgrundlage. Aber nicht immer, gerade in der frühen Phase ist es dann eher weniger eine Simulations Aufgabe bei uns als ein Konzept, wie so eine Maschine aussehen könnte, wie die Funktionalität realisiert wird und ähnliches.

[17:17] **Ich habe vorher rausgehört, wenn du jetzt von einem Produkt sprichst, dann ist es das Zusammenspiel, das ganze System, auch aus den verschiedenen Zukaufteilen der unterschiedlichen Hersteller.**

Ja. Am Ende machen wir XXXXXXX. Eigentlich in beiden Bereichen. Aber speziell bei den XXXXXXXXX ist es XXXXXXX, das heißt, wir verkaufen eine Komponente nur selten separat, eigentlich nie, sondern wir verkaufen sie immer als XXXXXXXX.
…

ROBUSTES DESIGN

[19:13] **Du hast vorher schon angedeutet, dass die Anforderungen wären des Entwicklungsprozesses leben. An der Stelle ist jetzt die Frage, zu welchen Problemen es führt im späteren Verlauf, wenn sich Anforderungen häufig ändern.**

Ich meine diese Robustheit, die erkauft man sich häufig mit zusätzlichen Kosten. Ein robustes Design heißt, für mich, in dem Kontext oft, dass wir was overengineeren. Denn wenn wir was an die Auslegungsgrenze designen, dann ist es klar, dass ich keinen Spielraum mehr habe, falls ich höhere Lastanforderungen habe. Robustes Design in dem Kontext würde für mich heißen, ich lege da mehr Material rein, ich mache es robuster auch für höhere Lasten. Und damit erhöhe ich aber auch die Kosten. Das heißt, es ist immer eine Abwägungssache. Aber natürlich ist es ein Thema, das so zu designen, dass man robust ist hinsichtlich dessen, wenn sich die Anforderungen ändern. Es ist ja nicht nur im Entwicklungsprozess, das ist ja auch so, dass sich die

Anforderungen der Kunden ändern. Dass man dann nicht sofort ein Problem kriegt. Das Ganze steht aber, wie gesagt, oftmals dem Gedanken der möglichst niedrigen Herstellkosten entgegen.

[20:48] **Siehst du da weiteren Handlungsbedarf derzeit bei euch im Unternehmen?**

Ich meine, da ist immer Handlungsbedarf da. Natürlich werden auch bei uns, oder treffe ich manchmal da falsche Entscheidungen und wenn da was unterstützen kann oder jemand, dann ist es immer gut.

[21:17] **Im Moment wird das also eher aufgrund von Erfahrungen designed?**

Ja.

KOMPLEXITÄT

[21:34] **Was verstehst du unter Komplexität bei euch? Was ist da die Herausforderung, wenn du dieses Stichwort hörst?**

Ja ich meine Komplexität, das ist ein Dauerbrenner. XXXXXXXXXXXXXX. Ich meine, Komplexität ist, wenn ich einen Hebel drücke und nicht weiß, was passiert.

Genau. Dazu habe ich auch eine konkrete Frage. Wie ist das bei euch, wenn ihr jetzt was verändert an eurer Maschine zum Beispiel ist es dann klar, wie sich das auf die Hersteller auswirkt oder wie dann andere Komponenten angepasst werden müssen?

Also ich würde grundsätzlich sagen, dass die Maschinen kompliziert sind aber nicht komplex. Es heißt, man hat es eigentlich im Griff, wenn man etwas verändert, was dann woanders passiert. Ja, natürlich, weil wir die Maschinen auch kennen.

Ich habe Komplexität jetzt im Sinne des Entwicklungsprozesses verstanden. Oder das ist das, wo ich eher Komplexität sehe.
Auf der Produktseite, das sind komplizierte Sachen. Da muss man sich lange reinlernen. Da gibt es Experten, die machen das seit Jahren, Jahrzehnten. Das ist alles mehr oder weniger handelbar, da gibt es manchmal ein technisches Problem, dass man lösen kann oder nicht. Ich würde es nicht als "komplex" bezeichnen.

Bei den Prozessen sieht's anders aus, weil es oftmals nicht klar ist, auch wenn man an andere Abteilung Aufgaben übergibt, was steht da eigentlich bei denen dahinter? Wie tun die das? Mit wem reden die? Dann kommt bei irgendjemand anderem was anderes an. Der fragt wieder bei der Entwicklung zurück.

[kurzes Verbindungsproblem.]

Also, wie gesagt, ich sehe es eher auf der Prozess Seite, dass man gerade wo Menschen auch miteinander interagieren, - wir geben irgendeiner Abteilung A irgendeinen Auftrag, einen Teil zu entwickeln, die Kosten zu erheben, was auch immer. Die sagen irgendjemand anderem, dass sie da was brauchen. Das kommt aber irgendwie anders an. Der fragt wieder zurück bei der Entwicklung. Und am Ende sind alle verwirrt…

Es ist nicht nur Kommunikation, das ist jetzt nur ein Beispiel. Aber auch da kommt dann oft Komplexität mit rein, weil dann auch Teilaufgaben übernommen werden müssen, wo man nicht richtig weiß, wie es eigentlich geht und dann wird es schnell in der eigenen Wahrnehmung komplex, weil man dann nicht mehr die Auswirkungen versteht von dem, was man tut.

Gut, danke. Wir machen dann mal an der Stelle Schluss und ich stelle dir mal unsere Forschungsansätze vor.
Das Variantenmanagement lassen wir mal weg. Ich nehme an, ihr habt in den XXXXXX nicht so viele Varianten.

Wir haben schon zahlreiche Varianten. Also ja, Variantenmanagement ist ein Thema, Gleichteile und so weiter. Dass es die Maschinen in unterschiedlichen Größen gibt und so weiter in der Ausführung.

[25:38] **Aber du hast es ans Ende gesetzt, weil...**

Ich sehe da kein großes Problem. Wir haben die Varianten. Der Baukasten funktioniert relativ gut. Man kann da immer was tun. Aber ich würde sagen, dass ist vom Reifegrad her angemessen für das, was wir brauchen. Das haben wir im Griff mehr oder weniger.

Dann belassen wir es mal dabei. Einer der Forschungsansätze beschäftigt sich auch mit der Produktfamilien Auslegung, da kommt es vielleicht auch nochmal zur Sprache.

…

TEIL 2

[36:29] **So, das war jetzt mal unsere Forschung im Schnelldurchlauf erklärt. Wir haben jetzt noch ca. zehn Minuten. Wo siehst du Verbesserungspotenzial oder Chancen und Stärken und welcher der beiden Ansätze könnte für euch interessant sein?**

Also bei dem automatisierten Modellerstellung mit Datenbanken habe ich nur nicht so ganz verstanden, wofür das gut ist. Wofür man das brauchen könnte.
...

Ok, also den Sinn dahinter verstehe ich, das ist ja letztendlich eine System Beschreibung. Im Sinne von was habe ich für Variablen und wenn ich an denen drehe, was passiert denn eigentlich. Es ist eigentlich ein Modell zur Beschreibung der Komplexität.

Genau, und um den Ansatz der Lösungsräume anwenden zu können, müsste man das sonst selbst aufbauen.

Ah, okay, stimmt, wir sind ja bei den Lösungsraum. Das heißt, wenn ich jetzt so ein Modell habe, dann geht es eigentlich darum, das Ganze durchrechnen zu können. Oder?

Genau. Es ist eine Hilfestellung für die Vorbereitung der Lösungsräume, wenn man so will.

Ja, ich sage mal eine automatisierte Modelldarstellung mit Datenbanken kann ich mir dahingehend vorstellbar, weil wir ja Systeme in verschiedenen Baugrößen haben zum Beispiel. Da wäre natürlich ein Wiederverwertungsgrad dar. Dass man sowas wieder nutzen kann.

Ob das dann wirklich automatisiert ist? Kann ich mir jetzt gerade nicht vorstellen, aber modularer Aufbau wäre da prinzipiell gut, weil man Sachen wiederverwenden kann. Jetzt gerade, wenn man einen Baukasten hat.

[41:25] **Du kannst es auch in einem größeren Abstraktionslevel sehen, dass ihr dann die verschiedenen Bausteine zum Beispiel hat. Die unterschiedlichen Geräte oder Zukaufteile könntet ihr mit euren Schnittstellen koppeln. Das ginge auch.**

Ja. Probleme in der Praxis, die man da dann hat.
Da wären ja auch Geräte von Lieferanten drin, mit verschiedenen, sagen wir mal, Größen was die bedienen können, sei es Spannend oder Last, was auch immer. Die verändern sich natürlich auch kontinuierlich. Da werden Gerät abgekündigte, werden Parameter verändert und so weiter. Das müsste in diesen Modellen auch alles mit getrackt werden.
Das heißt, da ist mit Sicherheit ein gewisser Pflegeaufwand notwendig. Heißt aber nicht, dass es das nicht wert ist, das sollte man nur grundsätzlich bedenken.

[42:25] **Ja, genau. Das ist ja genau der Sinn, dass wir auf der Suche nach Punkten sind, die wir bedenken müssen.**

Das geht in eine andere Richtung. Aber ich könnte mir sowas auch vorstellen, im Hinblick auf Auswertung von Fehlerfällen.
Dass der Ansatz da hilfreich ist, man kann ja zum Beispiel sagen. Also unsere Systeme sind weltweit im Einsatz und dann ruft ein Kunde an und sagt er hat folgendes Problem und dann kommen dann über die Jahre Hunderte von Anrufen.
Und diese Fehlermeldungen, die kann man dann Mappen. Bei jeder Fehlermeldung wird ja sozusagen ein Incident kreiert, dem geht der Service nach und am Ende kann man dann feststellen, folgende XXXX ist gebrochen, folgendes Bauteil ist kaputtgegangen und so weiter. Das heißt, da gibt es eigentlich ein Mapping, das man auswerten kann zwischen Fehleranzeigen und Fehlerursachen.

In dem Kontext könnte ich mir vorstellen, dass man da - Lösungsraum ist da jetzt das falsche Wort - aber einen Problemraum erstellen könnte, um zu sehen, wo die Probleme wirklich herkommen.

[43:54] **Ja, das ist auch ein interessanter Ansatz, ja.**

Das könnte wiederum helfen, um dann zu sagen: bei folgendem Fehlercode ist basierend auf diesem Modell das wir pflegen im Hintergrund höchstwahrscheinlich folgende Sache die Fehlerursache. Oder oftmals tauchen auch zwei Fehlermeldungen zusammen auf, was dann vielleicht dieses Modell auch auswerten kann hinsichtlich, wenn diese zwei Fehlermeldung zusammen auftauchen, dann liegt die Ursache wahrscheinlich in folgendem.

[44:39] **Wie sieht es mit dem anderen Ansatz aus, mit der Produktfamilienauslegung?**

Ja ich mein. Ich gehe davon aus, egal wen ihr befragt, wenn ihr sagt, damit können wir die Produktkosten optimieren. Da wird jetzt keiner sagen, nee interessiert mich nicht.

Jaaa, es gibt zumindest Unternehmen, die sagen, Varianten sind für sie uninteressant und teilweise geht es tatsächlich auch von der Firmenphilosophie her nicht darum, Kosten zu sparen, sondern den Entwicklern die technischen Möglichkeiten zur Verfügung zu stellen, die sie haben.

Ja. Klar. Aber es wird auch niemand aktiv nein sagen, wenn jemand Kosten sparen will. Von dem her ist es immer interessant.

[45:27] **Schaut ihr euch auch so Standardisierungen an?**

Klar, wir haben XXXXXX standardisiert und diverse andere Sachen. Das passiert aber heute meist darauf, dass man letztendlich sich aus dem ERP System rauszieht, was wie oft bestellt wurde im letzten Jahr, der letzten zwei Jahre, der letzten drei Jahre, um dann zu schauen wo ist eigentlich der größte Hebel?

Das heißt, da nimmt man eher die vorhandenen Ist-Daten. So machen es wir und schaut dann, wo man was standardisieren könnte.

Vieles Sachen liegen vielleicht auch auf der Hand. Viele Sachen liegen auch nicht auf der Hand. Die haben wir vielleicht nicht standardisiert, weil es uns noch nicht aufgefallen ist. Und das ist etwas, was man mit so einem systematischen Ansatz natürlich identifizieren könnte.

[46:31] **Und die Informationen über Kosten, unterschiedlicher Materialkosten oder wie sich Komplexitätskosten vielleicht auch sogar auswirken können auf die Entwicklung, ist das bekannt? Wird das transparent gemacht für die Entwickler?**

Muss man klar differenzieren.
Materialkosten, Herstellkosten des alles bekannt, dass ist auch alles festgelegt und kommt alles ins System. Manch einer hat da keinen Zugriff darauf. Aber diese Kosten sind alle da.

Komplexitätskosten und diese indirekten Kosten, die werden nicht direkt erfasst, nicht sind nicht bekannt.

[47:12] **Siehst du da ein Problem, dass sie nicht benannt werden? Oder denkst du, es könnte eine Möglichkeit geben, die besser zu dokumentieren?**

Ich wüsste nicht, wie man das in einem System machen kann, sodass Aufwand und Nutzen im Verhältnis stehen. Das ist der Punkt.
...

Das läuft dann eher auf der qualitativen Schiene ab, dahingehend, weil diese Auswahl an Stücklisten zu kompliziert sind, in der Abwicklung Fehler passieren, dann, irgendwo auf der Welt ein falsches Teil geliefert wird oder wir etwas Falsches bauen, das akzeptierter der Kunde nicht. Da entstehen dann natürlich Qualitätskosten. Die sind wieder messbar, die aber darauf zurückzuführen sind, dass bei uns was zu komplex ist und dadurch Fehler passieren.

Dann würde man da entsprechend dann die Stücklisten anpassen. Aber das ist rein auf der qualitativen Schiene.

AUSSAGEN UND HYPOTHESEN

[48:48] **Gut, dann werde ich dich jetzt abschließend noch einmal bitten, dass Du auf den vorherigen Link zurückgehst. Da sehen wir jetzt nochmal Aussagen und Hypothesen, die wir im Laufe der letzten Stunde besprochen haben. Bewertet die bitte mal von stimme überhaupt nicht zu, bis stimme völlig zu.**

...

Ich möchte anmerken, dass die drittletzte Frage so zugestellt ist, dass da immer eine positive Antwort rauskommt. Die ist so eigentlich unfair gestellt. Eine automatisierte Modell Erstellung verringert immer den Modellierungsaufwand, weil der Modellierungsaufwand ist die Modelldarstellung und wenn die automatisiert ist, dann ist es automatisch verringert.

Gut. Man könnte natürlich auch sagen, dass den Modellierungsaufwand, den wir betreiben, dafür brauchen wir keine Datenbank, das können wir händisch.

Dann muss ich trotzdem sagen, wenn es automatisiert wäre, wäre es schneller, weil ein Computer macht es immer schneller als händisch.

Ich seh das ein, das müssen wir bei der Auswertung beachten.

Gut fertig.

Gut, dann sind wir auch am Ende von meiner Seite. Danke nochmal für deine Teilnahme und die interessanten Einblicke.

Gerne.