27th INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2025
HOBOKEN, NJ, USA, 24 – 26 September, 2025

Methodology for Predictive Value Engineering in the Early Product Development Phases

Tarek Musa [1] [2]

[1]Liebherr Machines Bulle SA, Rue Hans-Liebherr 7, 1630 Bulle, Switzerland
[2]Technische Universität Ilmenau, Ehrenbergstraße 29, 98693 Ilmenau, Germany

Abstract: Mastering new product development at competitive costs is viewed as a challenge. Since the design process requires multiple iterations to fulfill the product's function in the early development phases, Value Engineering (VE) is used as a function-oriented approach. To address the challenge of overengineering, VE aims to improve the product while reducing costs without compromising quality. However, iterations during VE projects are often based on expert knowledge. This paper introduces a methodology to integrate predictive VE for new Product Generations in the context of Digital Twin technology. The methodology includes the creation of a VE knowledge base for qualitative VE prediction. To reach consensus on a Feature Engineering set, the collection of context-specific criteria is supported by the Delphi method. As a result of an informational Digital Twin, a hybrid Property Graph is presented to support quantitative VE prediction. Ultimately, the methodology is applied to an industrial case study.

Keywords: Value Engineering, Product Generation Engineering, Digital Twin, Design Theory, Graph Theory

1 Introduction

(Liu, 2020) notes that 85% of manufacturing costs occur during the early phases of product development (PD). As a primary reason for increasing product costs, overengineering is described as overspecification that does not necessarily provide more value to the customer. (Schlattmann and Seibel, 2024; VDI, 2011) To optimize product costs during PD, Value Engineering (VE) is used as a method to enhance the product's value by focusing on function-oriented costs. For example, the decision to perform VE in the early phases of PD is driven by the need for additional functions intended to increase the product's value. Further reasons to initiate VE are high product costs, insufficient functions, or lack of quality during the product lifecycle. To achieve this, product costs must be considered during early design iterations in PD for cost-efficient development of new products and to prevent overengineering. However, VE often relies on expert knowledge across various interfaces.

This paper presents a methodology to support the design team during the early stages of PD with predictive VE, explicitly aiming to avoid overengineering. The main contribution of this methodology is enabling qualitative VE prediction, focusing on context-specific criteria to guide function cost analysis when physical components are not yet available in the early PD. The paper presents the current state of VE based on literature, addresses related research challenges, and defines research objectives for predicting VE in the early phases of PD. The article analyzes the suitability of function-oriented design principles for predictive VE based on design theory approaches. Particularly when adding new functions in the context of Product Generation Engineering, the paper introduces a graph-based approach to specify the value-driven design. The methodology employs qualitative VE prediction based on context-specific knowledge to establish consensus on the boundary conditions of analyzed functions. Additionally, it integrates the Digital Twin technology by linking the knowledge base to the Digital Shadow of Product Generations, enabling quantitative VE prediction. Finally, the methodology is validated through an industrial case study, which facilitates subsequent quantitative VE predictions.

2 New Product Generation Development

According to (Ropohl, 2009), a product always consists of several elements and can be described as a system. A system is a model that can be described through constituent subsystems, which in turn comprise components. (Pahl et al., 2006) The product development aims to collect information about requirements and existing constraints following the clarification of an initiated task. Initially, the requirements list representing the customer needs is intended as a basis for the subsequent phases of conceptual and embodiment design during the early PD. In this Chapter, the early PD phases are introduced, followed by the essentials of Product Generation Engineering.

2.1 Early Product Development Phases

In the conceptual design, the principle solution is identified by abstracting the essential problems at the system level. Over time, the concretization process often includes preliminary materials or rough dimensions. During embodiment design, it becomes difficult or even impossible to modify the solution principle of the concept. At this stage, the design is developed into a more concrete draft that explores, for example, the function. Additionally, the design of individual parts is detailed, including dimensions, surface properties, and estimated costs (Pahl et al., 2006). (Ulrich and Eppinger, 2016) The final design is then finalized in the bill of materials (BoM).

2.2 Product Generation Engineering

According to (Albers et al., 2019), every PD is based on existing reference system elements (RSEs) as a basis for future development activities, such as subsystems from predecessors or competitors. (Albers et al., 2020a) In Product Generation Engineering (PGE), three variation types are differentiated to describe the development activities, e.g., concerning properties or functions of the new Product Generation Gn. A RSE can be transferred to Gn. However, it may also be adapted to boundary conditions while maintaining the solution principle, which is referred to as carry-over variation (CV). (Albers et al., 2020b) extended the term of embodiment variation due to the limited level of physical subsystems to attribute variation (AV) in the system context during development. The AV describes the modification of a new system element during development, maintaining a solution principle. The variation of solutions is described as principle variation (PV) when the solution principle is adapted. (Albers et al., 2022) generalized the description of results and realizations during the new development as System Generation Engineering (SGE).

3 Value Engineering

Overall, the design has a significant contribution to product costs in the early PD phases. Many decisions are needed to evaluate the design maturity based on specific requirements with limited cost information (Ehrlenspiel et al., 2020). To evaluate the design in terms of fulfilling needs at the least possible cost, VE is used as a method to decompose the product or its constituent parts into functions that need to be evaluated in relation to the customer's needs. The objective is to consider only functions that contribute to the customer value with appropriate specifications. The function analysis is described as the most important method for analyzing the effects on a product´s value. As a guiding step, the as-is functions are analyzed by gathering them through moderation with team members and compiling them into a function structure, such as a function tree, followed by the to-be functions. Typically, functions are classified as main functions, part functions, and subsequent functions. The VE process is initiated by identifying the market needs of the product. One of the key success factors of VE is interdisciplinary teamwork, which enables different designations of functions as a basis for structuring function dependencies and mapping costs accordingly. Based on the defined customer needs, the functions and related costs are analyzed. The value is created by deriving functions from the needs of new products (VDI, 2011).

One possible method to prioritize product requirements is the Kano model of customer satisfaction by (Kano, 1984; as cited by Sauerwein et al., 1996), which distinguishes between three types of identified requirements. Essentially, must-be requirements are defined as basic, which the customer already assumes. Not only do the basic requirements represent a competitive factor, but the customer would be highly dissatisfied if they are not fulfilled, and they would likely be uninterested in the product. Moreover, one-dimensional requirements describe customer satisfaction as proportional to the degree of fulfillment. This means that the customer´s satisfaction is increased if the fulfillment of the requirement is higher and vice versa. According to (Berger et al., 1993; as cited by Sauerwein et al., 1996), one-dimensional requirements are specified, measurable, and technical, which are commonly required by customers. In contrast, attractive requirements define the highest level of customer satisfaction and are not explicitly requested by the customer. The attractive requirements, as the third category of the Kano model, cause delight and lead to more than proportional customer satisfaction. Through the practical application of the Kano model, the classification of product requirements facilitates prioritization in product development. For instance, focusing on must-be requirements with most efforts may not be valuable, as customer satisfaction is often perceived more through the improvement of one-dimensional or attractive requirements (Sauerwein et al., 1996). As presented by (Marchthaler, 2008; cited in VDI, 2011), Kano is one of the various methods used in value analysis. (Ullah and Tamaki, 2011) also strengthen the Kano model, particularly in prioritizing features in product development to highlight customer satisfaction.

(VDI, 2011) After analyzing the functions, the creative phase is commonly initiated by collecting and developing ideas to perform the functions. Here, brainwriting and brainstorming methods are often used to drive creativity. For instance, the 635 or Brainpool, as brainwriting methods, aim to collect solution ideas from participants and exchange them between the participants to combine written ideas or trigger new ideas. The beneficial impact of commonly used creativity techniques in value analysis is the improvement in idea quality after multiple rounds. However, the goal of this paper is to collect context-specific knowledge for predictive VE before making decisions about design solutions. (Sekayi and Kennedy, 2017) strengthens the Delphi method as a qualitative approach to develop decision-making criteria through a series of defined rounds, gathering expert views towards a broader purpose. (Linstone and Turoff, 2002) The Delphi method is a survey technique used to reach consensus on future issues by utilizing structured expert questionnaires completed anonymously. The Delphi method also supports decision-making when historical data gathering is not available or not accurately known. In addition, (Mozuni and Jonas, 2017) emphasize the Delphi method for future-oriented design with context dependency.

3.1 Research Challenges in Value Engineering

In the literature, there are specific approaches that extend and combine VE, such as with target costing. (Bock et al., 2016) combined a decision model with target costing and pricing to support efficient product quality in VE. Because they focus on decision-making in sales and production departments, the proposed method does not explicitly mention the early PD.

Although it considers iterative loops to control variances, the prediction of VE, particularly regarding function costs, is not explicitly addressed. (Ibusuki et al., 2006) also combine VE and target costing in their approach, even focusing on function costs. While the early stages, such as the concept phase, are introduced, the prediction of VE is not explicitly addressed. Additionally, the related case study maps functions to physical components, detailing the associated costs. (Maisenbacher et al., 2013) developed an integrated VE (IVE) model by mapping components, functions, and requirements across three levels of a product. This model emphasizes comparing current and target costs at each level. (Maisenbacher et al., 2015) adapted the model for evaluating different jet engine concepts, highlighting the main domains of functions, components, and contact areas for concept assessment. Though the concept is discussed early on, the prediction of VE is not explicitly addressed in this context. Moreover, (Maisenbacher et al., 2016) highlight the importance of functions as a domain within the IVE model, providing initial insights for future concept development. They also distinguish between useful and harmful functions. However, the relationship between a function's importance and the early phases of PD is not explicitly discussed. Later, Behncke et al. (2014) expanded the IVE model by incorporating manufacturing processes and supply chain networks to provide additional information on physical components. (Sadi et al., 2015) also build on the IVE model by emphasizing information flow for early stakeholder analysis in the new PD. However, the prediction of VE is not explicitly addressed in their contribution.

Several research challenges for VE have been identified in the literature. Although the requirements, functions, and component costs are mapped to determine the importance of functions, the connection between functions and context-specific criteria in the early PD phases is not explicitly addressed. Based on the state of the art in VE (Musa et al., 2025) developed a concept for Digital Twin-based predictive VE to support design and purchasing in the early PD phases. The Digital Twin-based approach is initiated by the Physical Twin representing the context-specific VE knowledge, followed by the Digital Shadow as Phase 2 for historical Product Generation data, and Phase 3 as Digital Twin model, more specifically (Wilking et al., 2021) as an Informational Digital Twin (IDT) with a Digital Master for model-based VE prediction. Since physical components do not exist in the early PD phases, the main research problem for this paper is to develop a methodology for integrating predictive VE in the context of a Digital Twin.

3.2 Research Objectives for Predictive Value Engineering

Since physical components do not exist in the early phases of PD, a new method is needed to represent the relationship between functions and context-specific VE knowledge, incorporating properties for quantitative VE prediction. The following presents several function-oriented design approaches to investigate their suitability for addressing the research problem. Regarding design theory methods, various approaches highlight the understanding of design principles. According to (Hatchuel et al., 2004) the Concept-Knowledge (C-K) theory focuses on defining design based on concept and knowledge spaces. As the knowledge space transfers to concept space as the design square, the connection between C and K is emphasized as an expansion in the theory. Although the C-K theory is intended to support proven redesigns of the functional space, the goal of this paper is not fully met, since it does not explicitly represent functions within the context of customer needs with related properties for quantitative VE prediction. (Weber, 2005; as cited by Weber and Husung, 2016) developed a combined product and process modeling approach based on the distinction between characteristics and properties. This model is known as Characteristics-Properties Modeling (CPM) and Property-Driven Development (PDD), collectively referred to as CPM/PDD. The model considers both the synthesis and analysis of characteristics and properties. While designers can directly determine and influence product characteristics, the properties related to product behavior can only be indirectly affected by changing these characteristics. The CPM/PDD model also includes functions as properties, but the connection between functions and context-specific criteria for predictive VE is not explicitly highlighted. (Suh, 1998) introduced a design theory based on mapping customer, functional, physical, and process domains. Customer needs (CNs) are part of the customer domain, and the functional domain specifies these CNs as functional requirements (FRs). To meet these FRs, design parameters (DPs) are used in the physical domain. These DPs are then mapped to process variables (PVs) to produce the final product. Based on two main axioms, one for the independence of FRs and another for minimizing design information, the theory is called axiomatic design. Although Suh outlined the main domains necessary for VE, this approach does not fully address the research problem. Due to the axiomatic rule of independence, it cannot adequately map functions to context-specific criteria with related properties.

According to (Gruber, 1993), an explicit specification, such as for classes or functions, as represented in a knowledge base of a conceptualization, is regarded as an ontology. Furthermore, (Psyché et al., 2018) state that an ontology serves as an abstract model of a related world phenomenon, providing a structure for a domain, specifically as a basis for a knowledge base to explicitly specify the conceptualization. According to (Wilson, 2010), graphs typically consist of elements defined as nodes, which are connected through edges. (Sahlab et al., 2022) described ontology-based context models that correspond to a Digital Twin, representing knowledge and relationships specific to a particular domain. As a hybrid approach, a labeled property graph emerged as a combination of the ontology-based context model and the related database. The labeled property graph is described by a set of key-value pairs that represent the properties of two nodes and the relationships between them through edges. (Angles, 2018) emphasizes that a property graph represents specific characteristics of each node or edge as property-value pairs, resulting in a directed, labeled, and multigraph. In contrast, a property is demonstrated within the context of a property graph, for example, as a specific characteristic of relationships.

According to (Boudaoud et al., 2022), graph models facilitate the representation of complex data, for example, by mapping relational databases to property graphs. (Bonifati et al., 2022; Sakr et al., 2021) emphasize that property graphs represent the relevant abstractions required in future-oriented systems. Since the research objective regarding predictive VE addressed the relationship between functions and context-specific VE knowledge, with properties serving as specific features derived from customer needs, the property graph appears appropriate for labeling properties. Toward the primary goal of this paper, which is to predict VE qualitatively when physical components do not exist in the early PD and related feature data are not finally defined, the following research questions (RQs) emerge for this paper.

RQ1: How can context-specific criteria for functions be identified as function carriers for VE when physical components do not exist for new Product Generations in the early PD phases?

RQ2: How can an ontology be modelled to represent the relationship between context-specific VE criteria and functions in the early PD phases?

RQ3: How can the Product Generation data be modeled for context-specific criteria of the functions to develop an Informational Digital Twin-based predictive Value Engineering?

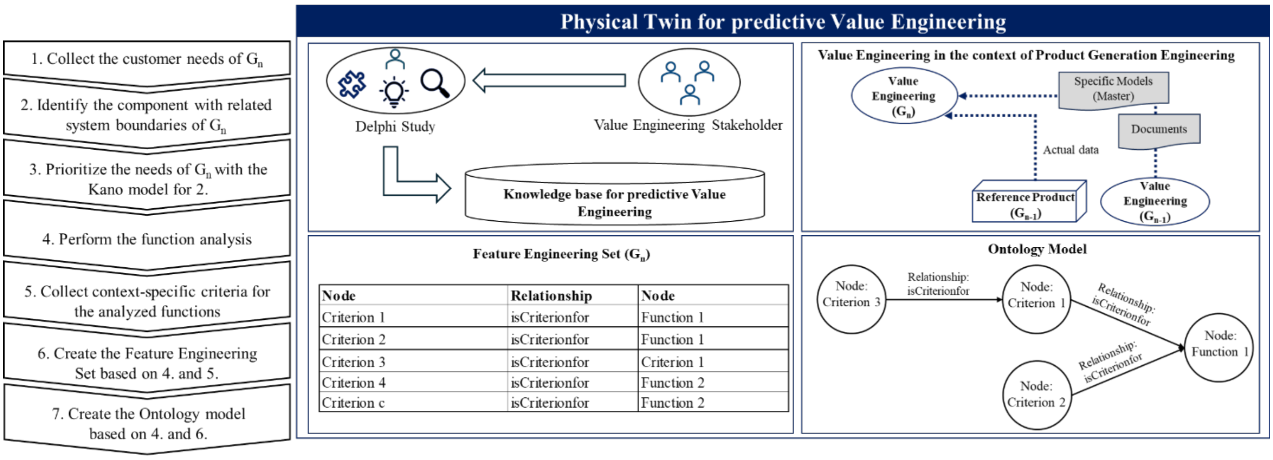
RQ4: How can a labeled Property Graph be modeled for an Informational Digital Twin-based predictive Value Engineering?

4 Methodology for Predictive Value Engineering

This Chapter presents a methodology to support the design in the early phases of PD toward predictive VE. Its goal is to guide the design team in Product Generation Engineering in making value-driven decisions. The methodology is structured in a step-by-step format, following the concept of Digital Twin-based predictive VE (Musa et al., 2025), as outlined in Chapter 3.1. The methodology focuses on Phases 1 to 3 to enable qualitative VE prediction by introducing the Informational Digital Twin (IDT) in Phase 3, which forms the basis for a Digital Master prediction model.

Phase 1: Physical Twin for Predictive Value Engineering

The goal of the Physical Twin as Phase 1 towards predictive VE is to identify context-specific VE knowledge for new Product Generations to prepare the related data acquisition in Phase 2. Since the scope is on predictive VE for a new Product Generation in the early phases of PD, the context-specific VE knowledge is not only based on existing Product Generation data. As previously presented in Chapter 3.2, one of the research objectives is not only to rely on historical physical components from databases for the function cost analysis, but also to collect context-specific criteria in the early PD phases associated to the functions as function carriers, as addressed by RQ1. To explore RQ1, the Physical Twin for predictive VE introduces the following methodological steps, as shown in Figure 1.



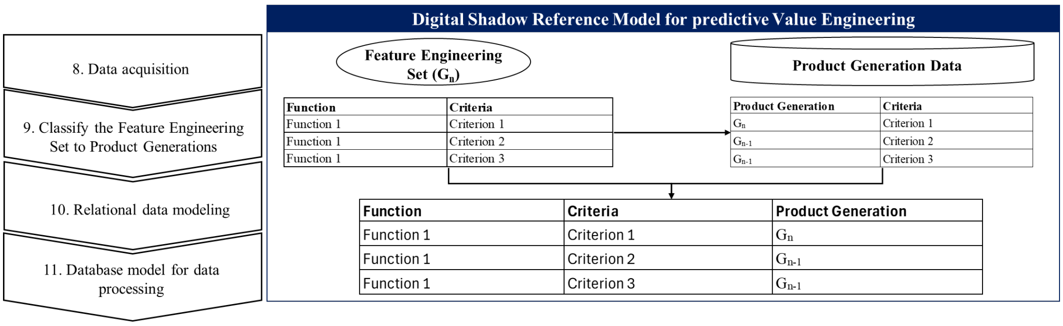
Node	Relationship	Node
Criterion 1	isCriterionfor	Function 1
Criterion 2	isCriterionfor	Function 1
Criterion 3	isCriterionfor	Criterion 1
Criterion 4	isCriterionfor	Function 2
Criterion c	isCriterionfor	Function 2

Figure 1. Physical Twin with methodology steps for Digital Twin-based predictive Value Engineering based on (Musa et al., 2025)

From a design perspective, as the initial step in the methodology, developing a new component begins with a new PD as a task (cf. Chapter 2). In this context, Gn represents the new Product Generation, and Gn-1 refers to the existing Product Generation (cf. Chapter 2.2). Based on the list of customer needs for Gn (cf. Chapter 2.1), the component for Gn is identified, along with its related system boundaries, as the second step. Depending on the early PD phase (cf. Chapter 2.1), defining the system boundaries for the new component is essential. For instance, if the new component is preliminarily defined with initial materials and rough dimensions during embodiment design, subsequent steps will rely on relevant context-specific criteria. The highest achievement of value-driven design for the new component may occur in the conceptual design phase, as several PVs need to be evaluated in accordance with the objectives and constraints of Gn (cf. Chapter 2.2). Suppose the component is in the embodiment design (cf. Chapter 2.1). In that case, the achievement of value-driven design may be influenced more than in the conceptual design, as the objectives are more concretized, and the solution principle may already be defined. The variation of product properties and functions, especially for VE, concerns the AV when modifying the RSEs (cf. Chapter 2.2). The focus of predictive VE is more for PV and AV rather than CV. In the case of detailing design, achieving value-driven design is less effective than in conceptual and embodiment design, as design details, such as dimensions, are essentially defined (cf. Chapter 2.1). However, the second step of the methodology emphasizes the vital importance of the early PD phase and clearly defined system boundaries. The later the development phase with a planned BoM release, the more it can impact the achievement of value-driven design as the objectives and constraints of Gn become more specified. As a third step, the needs of Gn are collected and prioritized from the requirements list (cf. step 1) as boundary conditions for the identified component (cf. step 2) by stakeholders. In particular, the methodology integrates the Delphi method to identify the interdisciplinary view on collecting needs, but as an anonymous expert questionnaire. Initially, an open question can be addressed to the stakeholder to determine which customer needs may be perceived as valuable from the development perspective of the new component. The third methodology step aims to prioritize requirements using the Kano model, as determined by stakeholders through the continuation of the Delphi method. The third methodology step is carried out through individual expert questionnaire rounds and subsequent consensus rounds, utilizing anonymous feedback. Subsequently, the function analysis of the new component is performed as the fourth step to analyze and model the dependency structure between the functions (cf. Chapter 3.1). The fifth step aims to collect context-specific criteria as boundary conditions for the analyzed functions. The goal is to define a knowledge base for predictive VE from the expert knowledge of historical Product Generations, focusing on early PD phases after achieving consensus. As a result of the collected context-specific criteria, the feature engineering set is created for Gn in step 6. The goal of step 6 is to classify prediction criteria for the data acquisition of Gn towards quantitative VE prediction. Furthermore, the prediction criteria need to be modeled to the analyzed functions of step 4, as addressed by RQ2 (cf. Chapter 3.2). Besides the exemplified feature engineering set in Figure 1, the related ontology is illustrated based on the direct relationship "isCriterionfor" between criteria 1 and 2, where function 1 serves as a node. Criterion 3 has a direct relationship with criterion 1 and an indirect relationship with function 1.

Phase 2: Digital Shadow Reference Model

Following RQ3 in Chapter 3.2, Phase 2 aims to develop a Digital Shadow reference model for predictive VE. Building on the feature engineering set classified in Phase 1, the related data are gathered in step 8 of Figure 4 from diverse data sources toward an IDT in Phase 3. The Product Generation data are generalized, which can be acquired as planning and development data, as well as actual data from existing Product Generations. In step 9, the feature engineering set (cf. step 6) is classified to the identified Product Generations. As illustrated in Function 1, criterion 1 is classified under Gn. For the existing Product Generations, e.g., Gn-1 and Gn-2, criteria 2 and 3 serve as classification input. Following step 10, the classified data are combined based on relational data modelling (cf. Chapter 3.2). As step 11, the acquired and classified data are processed to provide a database model for the IDT in Phase 3.



Function	Criteria
Function 1	Criterion 1
Function 1	Criterion 2
Function 1	Criterion 3

Product Generation	Criteria
G_n	Criterion 1
G_{n-1}	Criterion 2
G_{n-1}	Criterion 3

Function	Criteria	Product Generation
Function 1	Criterion 1	G_n
Function 1	Criterion 2	G_{n-1}
Function 1	Criterion 3	G_{n-1}

Figure 2. Digital Shadow with methodology steps for Digital Twin-based predictive Value Engineering based on (Musa et al., 2025)

Phase 3: Informational Digital Twin (IDT)

In Figure 5, the development of an IDT for predictive VE is illustrated with related methodology steps. In methodology step 12, the ontology model from step 7 is combined with the database model from step 11 to facilitate data processing of classified criteria to functions for existing and new Product Generations. As a result of this hybrid approach, a labeled property graph (cf. Chapter 3.2) is created for predictive VE in step 13. As illustrated by the following examples, Gn-1 and Gn, the labeled property graph contains, in addition to the exemplified ontology in Figure 3, the key-value pair as a property of the graph. For example, an existing Product Generation Gn-1 is mapped to a key-value pair identified from the processed database in step 11, which contains the context-specific criteria from step 5. In the case of a function as a node, the related key-value pair includes the definition of the analyzed function, e.g., generate power from step 4, and is provided in the feature engineering set of step 6. However, a further example is illustrated for the new Product Generation Gn, which contains for criterion 1 the key: NaN (Not a Number). Since criterion 1 is considered a missing graph property that cannot be identified from the database model in step 11, step 14 applies in the case where the key is NaN. The Digital Master-based prediction model focuses on the quantitative VE prediction of criterion 1, which may not be finally determined in the early PD phases. By combining the ontology with the processed database model of the Digital Shadow reference model, the labeled property graph aims to represent the IDT as a relationship between functions and their context-specific criteria, as well as between context-specific criteria and labeled values. In conclusion, Phase 3 aimed to address RQ4 by modeling a labeled property graph for predictive VE, with a particular focus on preparing the quantitative VE prediction of Gn.

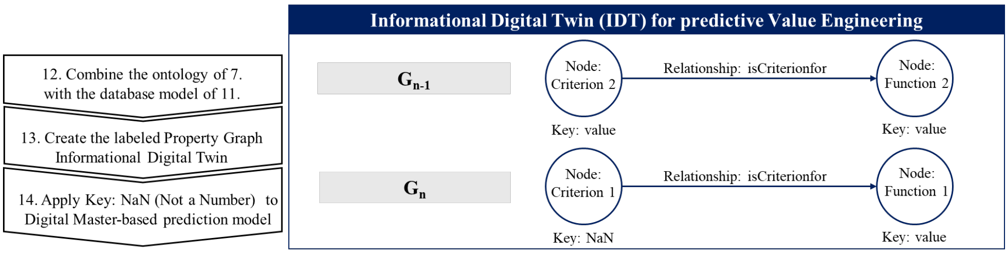


Figure 3. Labeled Property Graph for Digital Twin-based predictive Value Engineering based on (Musa et al., 2025)

5 Industrial Case Study

This Chapter presents an industrial case study of a combustion engine developer and manufacturer for large applications, such as Agriculture or Mining. The developed methodology in Chapter 4 is applied to the scenario of developing a new engine generation with a dry oil sump as a component for predictive VE. In Figure 4, the oil circuit subsystem is differentiated between the reference engine Gn-1, which has a wet oil sump system, and the new engine Gn, which has a dry oil sump system. The wet oil sump stores the entire engine oil, which is pumped and passes through the oil filter for lubrication throughout the engine. After the oil has circulated, it drains down, collected by the oil sump, and recirculates. In contrast, the dry oil sump stores most of the oil in a separate oil tank, allowing for more oil volume in the oil circuit. The oil drains down and is collected by the dry oil sump but is then pumped back to the oil tank by a scavenge pump. Not only does the design of a dry oil sump represent a new principle solution as PV, but new functions and properties may be considered as AV due to the influence of the scavenge pump and oil tank on the design of the dry oil sump. The oil pump and oil filter can be considered as RSEs, while the new system elements comprise the scavenge pump and oil tank. However, the scope of the case study is on the dry oil sump as a component according to methodology step 2.

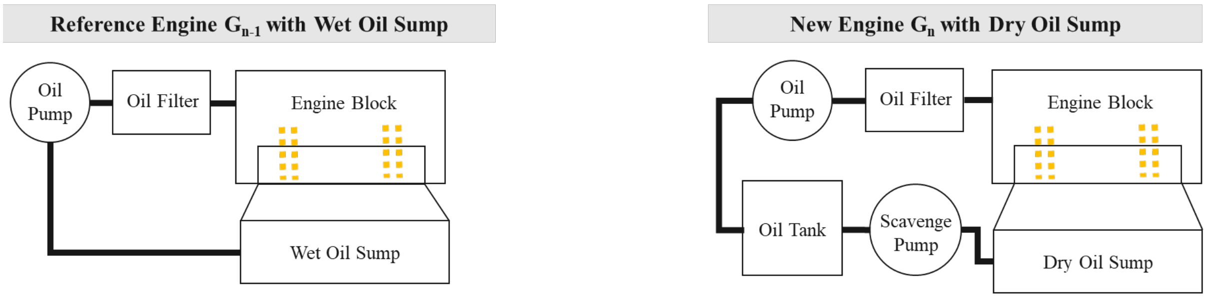


Figure 4. Principle solution of a wet oil sump system (Gn-1) and a dry oil sump system (Gn)

In Figure 5, the Physical Twin for predictive VE is shown for the dry oil sump. To understand which customer needs may be viewed as valuable for Gn from a development perspective, a Delphi study is conducted to reach consensus on the dry oil sump design based on expert knowledge. For the case study, a Senior Expert in Design at managerial level, a Head of Department for Simulation, a Subsystem Owner for the oil circuit, and a Design Engineer for the oil sump were chosen as stakeholders by a Value Engineer, who acted as the moderator. The selection of stakeholders is based on extensive experience in various new development projects and on representing multiple domains (system, subsystem, component). Using a requirements list, the current Delphi study is designed with an open question in the first round for the stakeholders. The aim is to gather customer needs individually that might be considered valuable for the oil sump design across different engine generations of base applications (cf. step 3). The second round seeks to achieve consensus on the oil sump design through anonymous feedback from each domain. Then, the needs are prioritized with the Kano model in the third round. Below, the stakeholders' views on illustrated needs are presented.

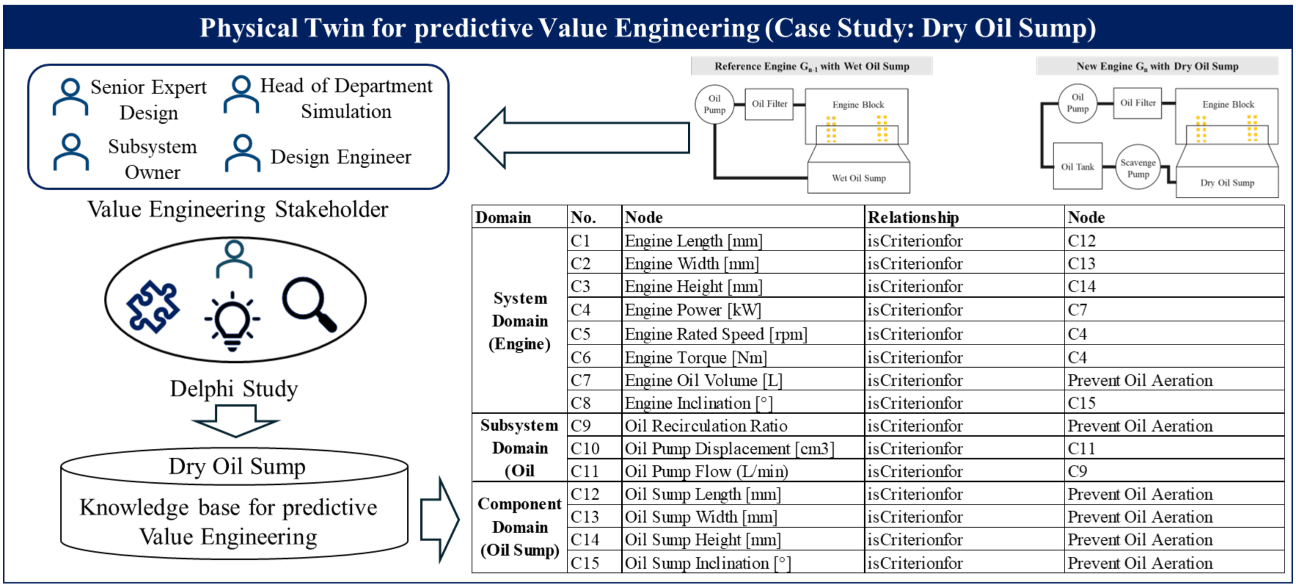
From the Senior Expert Design perspective, the achievement of criteria such as customer packaging that includes engine dimensions (length, width, height) or engine service interval was emphasized for management reporting. It is crucial to investigate possible configurations for the customer applications, which may require multiple iterations. By exchanging anonymous feedback on the oil sump design as a domain, the screw dimensions were evaluated as a basic requirement in the third round. According to the Senior Expert Design, a customer's need is classified as basic when it is non-negotiable. The contribution of the screws to the oil sump is towards maintainability and assembly. Regarding the engine inclination, the need has been classified as basic, as it is considered a prerequisite for the customer. As a relevant application, the center of gravity and construction of an excavator were highlighted due to the slope. The engine dimensions were classified as one-dimensional requirements, specifically regarding packaging compatibility with the customer's application. If the required engine dimensions for the customer's packaging are achieved more compactly, customer satisfaction is directly proportional to expectations. In contrast, the engine service interval from the first round is evaluated as an attractive requirement, which delights the customer in terms of unexpectedly longer service intervals for the engine maintenance due to appropriate oil lubrication.

From the Head of Department for Simulation's perspective, the engine dimensions from round 1 have been highlighted for integration into the customer application. Furthermore, the engine's rated speed has been identified as a valuable need for the oil sump design. During feedback exchange on the oil sump domain in round 2, a consensus was reached regarding the oil sump dimensions. The engine dimensions, as criteria for the oil sump, have been classified as both basic and related to customer integration expectations. To avoid high part costs, a consensus was also reached on the oil pump criteria within the oil circuit domain. Consequently, an additional pump would be necessary if the oil sump is too flat and hinders proper oil pumping. In contrast to the evaluation of the oil sump design domain, the oil volume has been assessed as a performance requirement for the oil sump. Since the oil sump inclination (front, back, left, right) helps prevent oil aeration, ensuring the necessary oil volume is critical for the design. Additionally, the engine rated speed has been classified as a one-dimensional requirement in round 3. If the rated speed exceeds expectations, engine component simulations could improve brake performance. Conversely, downspeeding would help the customer save more fuel. Engine power is another performance requirement. However, higher power does not necessarily lead to increased customer satisfaction. If the power is higher, the customer may be satisfied but may not be willing to pay more. Moreover, engine lifetime and service interval have been evaluated as attractive requirements. The oil sump can help extend service intervals and engine life by providing better lubrication, as long as the oil volume requirements are satisfied by the oil sump design.

From the Subsystem Owner's perspective, engine dimensions were also emphasized as essential for integrating into the customer application. During feedback exchanges with the oil sump domain, a consensus was reached on the oil sump dimensions, classified as basic, in preparation for expected customer integration. In the oil circuit domain, the oil pump criteria were emphasized as boundary conditions for designing the oil sump. The oil recirculation ratio, which indicates how long the oil stays in the sump, was specified as a one-dimensional requirement, as it initially influences the sump size. Additionally, the flow and displacement of the oil pump are considered in terms of performance needs, as a small oil pump may not be able to handle a large volume of oil, which could potentially affect the oil sump dimensions. The required oil volume was considered a basic need for the oil sump design, as customer satisfaction could be very low if the engine is not adequately lubricated. Furthermore, the engine lifetime and service interval were evaluated as attractive. The oil sump can contribute to longer service intervals and engine lifetime by ensuring better lubrication if it provides the necessary oil volume. For example, in one application, the service interval was set as a boundary condition. However, after validation tests, a longer interval was identified as potentially beneficial.

From the Design Engineer's perspective, the oil sump dimensions have been assessed as a basic requirement for packaging compatibility. The dimensions of the screws were also categorized as basic to ensure better maintainability and prevent leakage. Although the screws are considered necessary, they are not a priority because the customer is unwilling to pay extra. Regarding the engine inclination, a consensus has been reached based on anonymous feedback, evaluating the oil sump inclination as a one-dimensional requirement. If integrating the engine into the customer application with an appropriately inclined oil sump allows for a more compact design, it could lead to higher customer satisfaction. By analyzing anonymous feedback from the oil circuit domain, the pump displacement and flow rate have been evaluated as basic by the oil sump designer. To move a specific volume of oil from the sump, the appropriate oil pump criteria are likely to be treated as boundary conditions in the design of the oil sump. Ensuring proper pumping while preventing oil aeration requires considering the shape of the oil sump. Without draft angles, the shape approximation during early design phases may become more complex. A lack of initial oil sump dimensions or inclination data could result in increased part costs due to the selection of a premium supplier for specific manufacturing. Nonetheless, the engine's lifetime and service interval are viewed as attractive requirements.

The function analysis was conducted with the stakeholders, following the 'how and why' principle (cf. step 4). The main function of the oil sump is to provide oil for engine lubrication. The as-is function structure of the wet oil sump includes the part function of storing oil. Collecting drained oil from the engine is considered as an additional function. To support customer maintenance, including oil changes, the function of removing dirt and oil must be incorporated into the oil sump design. To maintain the oil level, a subsequent function derived from the part function involves pumping oil from the sump. To prevent oil loss, it is crucial to maintain the material properties to avoid cracks. Ensuring the oil volume in the sump involves analyzing the function of tightening the sump bolts to secure the connection to the crankcase. Additionally, the function of preventing leaks and oil loss was identified through proper positioning and compression of the seal. To avoid direct contact with rotating parts, such as the crankshaft, the function of preventing oil splashing is linked to preventing oil foaming by shaping the oil sump to reduce oil movement. The to-be function structure of the dry oil sump differs slightly from the wet sump, as the dry sump primarily collects drained oil, while the oil tank primarily stores most of the oil as its main function. However, an additional function as AV for the dry oil sump was also analyzed to prevent oil aeration caused by the scavenge pump, which may pump oil with air. The effect of pumping oil with air is considered disadvantageous for the customer, as it may lead to engine damage due to reduced lubrication. Figure 5 illustrates the feature engineering set related to preventing oil aeration (cf. steps 5 and 6).



Domain	No.	Node	Relationship	Node
System Domain (Engine)	C1	Engine Length [mm]	isCriterionfor	C12
	C2	Engine Width [mm]	isCriterionfor	C13
	C3	Engine Height [mm]	isCriterionfor	C14
	C4	Engine Power [kW]	isCriterionfor	C7
	C5	Engine Rated Speed [rpm]	isCriterionfor	C4
	C6	Engine Torque [Nm]	isCriterionfor	C4
	C7	Engine Oil Volume [L]	isCriterionfor	Prevent Oil Aeration
	C8	Engine Inclination [°]	isCriterionfor	C15
Subsystem Domain (Oil	C9	Oil Recirculation Ratio	isCriterionfor	Prevent Oil Aeration
	C10	Oil Pump Displacement [cm3]	isCriterionfor	C11
	C11	Oil Pump Flow (L/min)	isCriterionfor	C9
Component Domain (Oil Sump)	C12	Oil Sump Length [mm]	isCriterionfor	Prevent Oil Aeration
	C13	Oil Sump Width [mm]	isCriterionfor	Prevent Oil Aeration
	C14	Oil Sump Height [mm]	isCriterionfor	Prevent Oil Aeration
	C15	Oil Sump Inclination [°]	isCriterionfor	Prevent Oil Aeration

Figure 5. Physical Twin for Predictive Value Engineering for an exemplified function of the dry oil sump

Based on the collected context-specific criteria for the function (Prevent Oil Aeration), which has both direct and indirect relationships, the ontology has been modeled (cf. step 7). According to step 8, data related to the feature engineering set has been obtained from PLM databases and classified to Gn (cf. step 9). Based on the relational data modeling of the PLM lists (cf. step 10), the relational database (cf. step 11) has been integrated into Neo4j for graph-based representation (cf. step 12). To understand which context-specific criteria are important for the additional function in a value-driven design of the dry oil sump, Figure 6 illustrates the labeled property graph (cf. step 13). For the case study, fictitious values were used to create the labeled property graph (see step 13), as shown in Figure 6. To highlight the importance of the additional function (Prevent Oil Aeration) as AV, the relationships have been modeled accordingly. For example, engine power (C4) is related to oil volume (C7), since higher engine performance requires more oil for lubrication. In total, 12 functions were evaluated with the stakeholders. Preventing oil aeration is a key function of the dry oil sump. For comparison, the function to position the seal was evaluated with fewer relationships, such as engine dimensions and oil volume, as direct relationships.

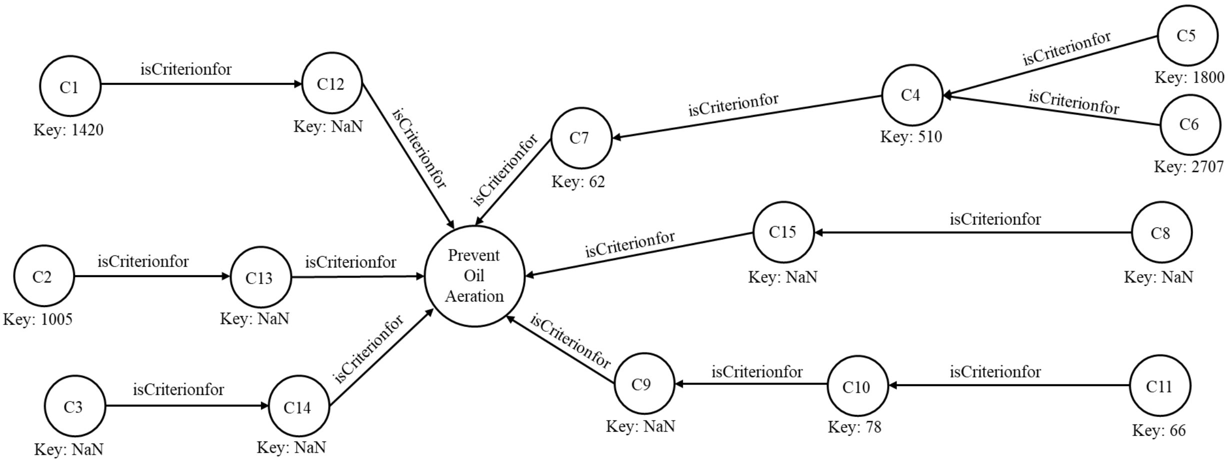


Figure 6. Labeled Property Graph for an exemplified function of the dry oil sump

In summary, incorporating the Delphi method into the methodology was seen as advantageous because it facilitated the exchange of insights across multiple domains in prioritizing customer needs for the oil sump design. The second round of the Delphi questionnaire, which included anonymous feedback, was considered subjective because not all stakeholders shared the same opinions on the oil sump design. However, it helped the participants in understanding the broader, context-specific criteria for the oil sump. Specifically, the methodology provided predictive support, helping the development team avoid overspecification by allocating the context-specific criteria to functions with direct and indirect relationships. Additionally, the Kano model is considered advantageous for prioritizing needs early in the project. For example, substantial effort might be directed toward a function that the customer does not value as highly, such as criteria like serviceability. Classifying needs using the Kano model was considered somewhat to quite challenging due to the qualitative abstraction of customer satisfaction into expectation levels. Notably, the methodology has been identified as a helpful guide during early PD to prevent misunderstandings related to the specifications. For example, defining design criteria such as screw dimensions or surface roughness without first linking them to specific functions can cause multiple iterations. For performance needs such as engine rated speed, assessments often rely on experience-based knowledge, which may be limited with a new engine generation. The methodology is valued as a predictive decision-support, especially when reference engines are not directly applicable to new design principles and their boundary conditions.

6 Summary and Outlook

Many design decisions are becoming more dependent on the early stages of product development (PD). Using Value Engineering (VE) as a method highlights the potential to reduce costs while avoiding overspecifications early in the PD process. To improve the quality of VE outcomes during these early PD phases, this paper introduced a methodology focused on qualitative VE prediction for new Product Generations. This methodology is developed within a Digital Twin-based predictive VE approach, which includes the Physical Twin in Phase 1 to build a knowledge base for predictive VE across multiple domains. The approach guides the initial phase by collecting context-specific criteria related to the functions of new components, supported by the Delphi method to reach consensus among various Product Generations. Additionally, the methodology is validated through an industrial case study that highlights the importance of VE in the context of Product Generation Engineering, examining a new principle solution. The importance of the function is supported by allocating and prioritizing customer needs using the Kano model, which in turn leads to a feature engineering set that results in an ontology model for predictive VE. The integration with a database model from the Digital Shadow reference model creates a graph-based representation of both direct and indirect relationships between context-specific criteria and functions. Therefore, the hybrid property graph aims to support subsequent quantitative VE predictions. The developed methodology has proven to guide the development team with value-driven decisions, which are crucial for later component purchasing in lifecycle phases. Based on this qualitative VE prediction approach, future research should explore methods for predicting missing values using historical Product Generation data. Additionally, the influence of predicted variation shares on function-oriented costs in the early phases of PD warrants further investigation.

Acknowledgments

This work has been supported by Liebherr Machines Bulle SA. The author sincerely appreciates the opportunity to conduct a case study and the participation of the stakeholders.

References

Albers A., Fahl J., Hirschter, T., Endl M., Ewert R., Rapp S., 2020a. Model of PGE – Product Generation Engineering by the Example of Autonomous Driving, 30th CIRP Design 2020, Procedia CIRP 91, pp. 665–677. https://doi.org/10.1016/j.procir.2020.03.113

Albers A., Rapp S., Fahl J., Hirschter T., Revfi S., Schulz M., Stürmlinger T., Spadinger M. 2020b. Proposing a generalized description of variations in different types of systems by the model of PGE – Product Generation Engineering, 16th International Design Conference, pp. 2235–2244. https://doi.org/10.1017/dsd.2020.315

Albers A., Rapp S., Spadinger M., Richter T., et al., 2019. The reference system in the model of PGE: Proposing a generalized description of reference products and their interrelations. ICED 19, Cambridge University Press, pp. 1693–1702. https://doi.org/10.1017/dsi.2019.175

Albers, A., Rapp, S., 2022. Model of SGE: System generation engineering as basis for structured planning and management of development, in: Krause, D., Heyden, E. (Eds.), Design Methodology for Future Products: Data Driven, Agile and Flexible, Springer, Cham, pp. 27–46. https://doi.org/10.1007/978-3-030-78368-6_2

Angles, R., 2018. The property graph database model, in: D. Olteanu, B. Poblete (Eds.), Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW 2018), Cali, Colombia, 21–25 May 2018.

Behncke, F.G., Maisenbacher, S., Maurer, M., 2014. Extended model for integrated value engineering, Procedia Computer Science, 28, pp.781–788. https://doi.org/10.1016/j.procs.2014.03.093

Bock, S., Pütz, M., 2017. Implementing Value Engineering based on a multidimensional quality-oriented control calculus within a Target Costing and Target Pricing approach, International Journal of Production Economics 183, 146–158. https://doi.org/10.1016/j.ijpe.2016.09.007

Bonifati, A., Dumbrava, S., Martinez, E., et al., 2022. DiscoPG: Property graph schema discovery and exploration, Proceedings of the 48th International Conference on Very Large Data Bases (VLDB 2022), Sydney, Australia, pp.3654–3657. https://doi.org/10.14778/3554821.3554867

Boudaoud, A., Mahfoud, H., Chikh, A., 2022. Towards a complete direct mapping from relational databases to property graphs, Model and Data Engineering (MEDI 2022), Vol. 13761, pp. 222–235. https://doi.org/10.1007/978-3-031-21595-7_16

Ehrlenspiel, K., Kiewert, A., Lindemann, U., Mörtl, M., 2020. Kostengünstig Entwickeln und Konstruieren. Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-62591-0

Ehrlinger, L., Wöß, W., 2016. Towards a definition of knowledge graphs, Institute for Application Oriented Knowledge Processing (FAW), Johannes Kepler University Linz, Austria.

Gruber, T., 1993. A translational approach to portable ontologies, Knowledge Acquisition 5 (2), 199–220. https://doi.org/10.1006/knac.1993.1008

Hatchuel, A., Le Masson, P., Weil, B., 2004. C-K Theory in Practice: Lessons from Industrial Applications, DS 32: Proceedings of Design 2024, 8th International Design Conference, Dubrovnik, Croatia, pp. 245-258.

Ibusuki, U., Kaminski, P.C., 2007. Product development process with focus on value engineering and target-costing: A case study in an automotive company, International Journal of Production Economics 105 (2), 459-474. https://doi.org/10.1016/j.ijpe.2005.08.009

Linstone, H.A. and Turoff, M., 2002. The Delphi method: Techniques and applications, New Jersey Institute of Technology, Original work published 1975. https://doi.org/10.2307/3150755

Liu, X., 2022. Cost-Engineering-System in den produzierenden Unternehmen. Methoden, Prozesse und Erfahrungsberichte aus der Praxis. Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-65096-7

Maisenbacher, S., Behncke, F.G.H., Lindemann, U., 2013. Model for Integrated Value Engineering, IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bangkok, Thailand, pp.1288–1292. https://doi.org/10.1109/IEEM.2013.6962618

Maisenbacher, S., Stanglmeier, M., Behncke, F., 2015. Integrated Value Engineering – adapted approach to assess different concepts of a jet engine, 9th Annual IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, pp.263–268. http://doi.org/10.1109/SYSCON.2015.7116762

Maisenbacher, S., Behncke, F.G.H., Roth, M., Fleckenstein, F., Roos, R., 2016. Integrated value engineering — Increasing the value of a forklift subsystem, Annual IEEE Systems Conference (SysCon), Orlando, FL, USA, pp. 1–6. https://doi.org/10.1109/SYSCON.2016.7490654

Mozuni, M., Jonas, W., 2018. An introduction to the morphological Delphi method for design: A tool for future-oriented design research, The Journal of Design, Economics, and Innovation 3(4), 303–318. https://doi.org/10.1016/j.sheji.2018.02.004

Musa, T., Rigozzi, A., Bignion, E., Pelloquin, P., Husung, S., 2025. Concept for Digital Twin-based predictive Value Engineering. Proceedings of the 35th CIRP Design, Patras, Greece [In press].

Pahl, G., Beitz, W., Feldhusen, J., 2006. Engineering Design: A Systematic Approach. 3rd ed., Springer-Verlag London Limited. http://dx.doi.org/10.1007/978-1-84628-319-2

Psyché, V., Anjou, C., Fenani, W., Bourdeau, J., Forissier, T., Nkambou, R., 2018. Ontology-based context modelling for designing a context-aware calculator, Workshop Proceedings of the 14th International Conference on Intelligent Tutoring Systems (ITS 2018), Montreal, Canada, 11–15 June 2018, pp.13–22.

Ropohl, G., 2009. Allgemeine Technologie: Eine Systemtheorie der Technik. http://doi.org/10.26530/OAPEN_422388

Sadi, T., Behncke, F.G.H., Maisenbacher, S., Kremer, S., 2015. Integrated Value Engineering – framework for the application of methods for visualization of information, Proceedings of the 20th International Conference on Engineering Design, ICED15, Politecnico Di Milano, Italy.

Sahlab, N., Braun, D., Köhler, C., Jazdi, N., 2022. Extending the intelligent digital twin with a context modeling service: A decision support use case, Procedia CIRP 107 (9), pp.463–468. https://doi.org/10.1016/j.procir.2022.05.009

Sakr, S., Bonifati, A., Voigt, H., et al., 2021. The Future is Big Graphs: a Community View on Graph Processing Systems, Communications of the ACM 64 (9), 62–71. https://doi.org/10.1145/3434642

Sauerwein, E., Bailom, F., Matzler, K., & Hinterhuber, H. H., 1996. The Kano model: How to delight your customers, International Working Seminar on Production Economics, Innsbruck/Igls, Austria, pp. 313–327.

Schlattmann, J., Seibel, A., 2024. Produktentwicklungsprojekte – Aufbau, Ablauf und Organisation. Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-67988-3

Sekayi, D., Kennedy, A., 2017. Qualitative Delphi method: A four round process with a worked example. The Qualitative Report, 22(10), pp.2755–2763. https://doi.org/10.46743/2160-3715/2017.2974

Sakr, S., Bonifati, A., Voigt, H. et al., 2021. The Future is Big Graphs: A Community View on Graph Processing Systems, CACM 64 (9), pp. 62–71. https://doi.org/10.1145/3434642

Suh, N.P., 1998. Axiomatic Design Theory for Systems, Research in Engineering Design (10), 189–209, Springer-Verlag London Limited. https://doi.org/10.1007/s001639870001

Ulrich, K.T., Eppinger, S.D., 2016. Product Design and Development, (6th ed.). McGraw-Hill Education.

Ullah, S. A.M.M., Tamaki, J., 2011. Analysis of Kano-model-based customer needs for product development, The Journal of The International Council on Systems Engineering 14 (2), 154–172. https://doi.org/10.1002/sys.20168

VDI, 2011. Wertanalyse – das Tool im Value Management. Springer Berlin, Heidelberg, https://doi.org/10.1007/978-3-540-79517-9

Weber, C., Husung, S., 2016. Solution patterns – their role in innovation, practice and education, 14th International Design Conference DESIGN 2016, Design Theory and Research Methods, Cavtat, Dubrovnik, Croatia, pp.99–108.

Wilking, F., Schleich, B., Wartzack, S., 2021. Digital twins – definitions, classes and business scenarios for different industry sectors, Proceedings of the 23rd International Conference on Engineering Design, ICED21, Gothenburg, Sweden, 16–20 Aug 2021. https://doi.org/10.1017/pds.2021.129

Wilson, R.J., 2010. Introduction to Graph Theory, 5th ed., Pearson Education Limited, Harlow, England.

Contact: T. Musa, Liebherr Machines Bulle SA, Rue Hans-Liebherr 7, 1630 Bulle, Switzerland, tarek.musa@tu-ilmenau.de.