

Methods

Iris Graessler* and Julian Hentze

The new V-Model of VDI 2206 and its validation

Das Neue V-Modell der VDI 2206 und seine Validierung

<https://doi.org/10.1515/auto-2020-0015>

Received March 6, 2020; accepted March 11, 2020

Abstract: Since 2016, a new version of the VDI (German Association of Engineers) Guideline 2206 has been developed by the Technical Committee VDI GMA 4.10 “Interdisciplinary Product Creation”. This article presents the revision results of the VDI Guideline 2206:2004 “Design methodology for mechatronic systems”. The core content of the guideline is an updated and enhanced V-Model for Mechatronic and Cyber-Physical Systems. The inherent concern logic of the V-Model represents the logical sequence of tasks. Its key advantage lies in staying independent from the chosen form of project organization. This way, the V-Model can be applied in classically managed projects as well as in agile projects. In addition, the article describes how the revision was performed and which potentials were tapped. Based on the identified potentials, the new V-Model is derived, explained and illustrated. New contents such as the introduction of checkpoints and the integration of requirements engineering are explained in detail. Furthermore, the pursued scientific procedure and the results of the International Validation Workshop with 25 experts from science and industry are proposed.

Keywords: V-Model, mechatronic and cyber-physical systems, interdisciplinary product creation

Zusammenfassung: Seit 2016 aktualisiert der VDI GMA Fachausschuss 4.10 „Interdisziplinäre Produktentstehung“ die VDI-Richtlinie 2206: 2004. Dieser Beitrag enthält die Überarbeitungsergebnisse der VDI-Richtlinie 2206 „Entwurfsmethodik für mechatronische Systeme“. Der Kerninhalt der Richtlinie ist ein aktualisiertes und erweitertes V-Modell für mechatronische und Cyber-Physische Systeme. Die inhärente Ablauflogik des V-Modells repräsentiert die logische Abfolge von Aufgaben. Ihr Haupt-

vorteil besteht darin, in der Darstellung unabhängig von der gewählten Form der Projektorganisation zu bleiben. Auf diese Weise kann das V-Modell sowohl in klassisch organisierten als auch in agilen Projekten angewendet werden. Der Beitrag beschreibt das wissenschaftliche Vorgehen und welche Potenziale ausgeschöpft wurden. Basierend auf den identifizierten Potenzialen wird das neue V-Modell abgeleitet, erklärt und illustriert. Neue Inhalte wie die Einführung von Kontrollpunkten und die Integration von Anforderungsentwicklung werden ausführlich erläutert. Darüber hinaus werden die Ergebnisse des Internationalen Validierungsworkshops mit 25 Experten aus Wissenschaft und Industrie dargestellt.

Schlagwörter: V-Modell, mechatronische und Cyber-Physische Systeme, Interdisziplinäre Produktentstehung

1 Introduction and motivation

Mechatronic products, formed by the disciplines mechanics, electrics/electronics and software are pioneers in using interdisciplinary procedures. In addition, Systems Engineering provides generic approaches and guidelines for the development of interdisciplinary products [1–5]. The first release of the VDI Guideline 2206 “Design methodology for mechatronic systems” of the German Association of Engineers (VDI), was published in 2004 [6]. Describing the procedure for the development of mechatronic systems, the V-Model for the first time was transferred from Software Engineering [7] to Mechatronics. The V-Model, which describes the macrocycle of product creation, represents the idea of interlinking all disciplines involved in engineering tasks. In order to meet the challenges of Cyber Physical Systems, Systems Engineering and Digital Business Models, the V-Model meanwhile had to be enhanced. Due to this need for revision, updating and extension of the V-Model for Mechatronic and Cyber Physical Systems, the VDI founded the Technical Committee (TC) VDI GMA 4.10 “Interdisciplinary Product Creation” in March 2016. This Technical Committee is chaired by the first author. The aim was to identify necessary changes, updates and revisions of the existing VDI 2206:2004 Guideline and to

*Corresponding author: Iris Graessler, Chair for Product Creation, Heinz Nixdorf Institut, Paderborn University, Paderborn, Germany, e-mail: iris.graessler@hni.upb.de

Julian Hentze, Dr. August Oetker Nahrungsmittel KG, Bielefeld, Germany, e-mail: julian.hentze@oetker.com

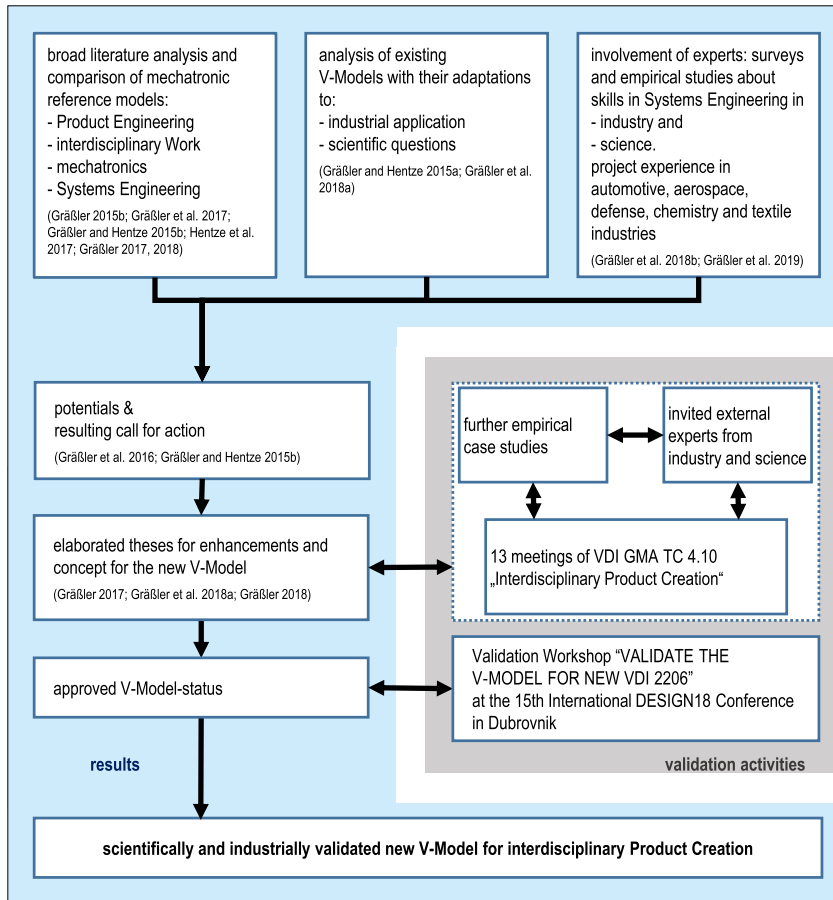


Figure 1: Scientific approach of VDI 2206 revision.

create and validate a new enhanced V-Model. In order to achieve excellent applicability in industrial practice, the TC is constituted by as many industrial as academic members. Industrial members cover the application domains automotive, automotive electronics, aerospace, defense, chemistry, woodworking and textile industries. The academic part is represented by scientists of the VDI society for measurement and automation technology as well as the VDI society for Product and Process Design. These include scientific employees working on their PhD theses as well as their instructing professors [8]. This structure guarantees fast results and deep discussions conducted from different fields of expertise. In the course of the committee meetings, additional experts for dedicated topics were invited to the TC meetings in order to even deeper and broader insights. The starting point for this publication is the motivation and call for action for changing the V-Model and the VDI 2206:2004 Guideline. Further, the chosen scientific approach is explained and validated results are summarized. Finally, the new V-Model is proposed and explained.

2 Scientific approach

The scientific procedure is characterized by a continuous interplay of results generation and in-process-validation. In order to identify potentials of enhancing the V-Model, the authors started their scientific procedure with three parallel analyses (Figure 1). First, literature dealing with topics like Product Engineering, interdisciplinary work, Mechatronics, Cyber Physical Systems, Systems Engineering etc. was analyzed and existing reference models were compared with each other [8–13]. Second, already existing V-Models and their individual adaptations to industrial and scientific application cases were analyzed. The individually tailored V-Models provide information on reasonable and already in numerous cases contextually modified aspects [14, 15]. Third, further experts were involved and project experience from different application domains was taken into account [5, 16]. As a consequence, potentials of enhancing the V-Model and the resulting call for action were derived [10, 17]. Thus, the call for action was worked out, examined, changed and adapted to the

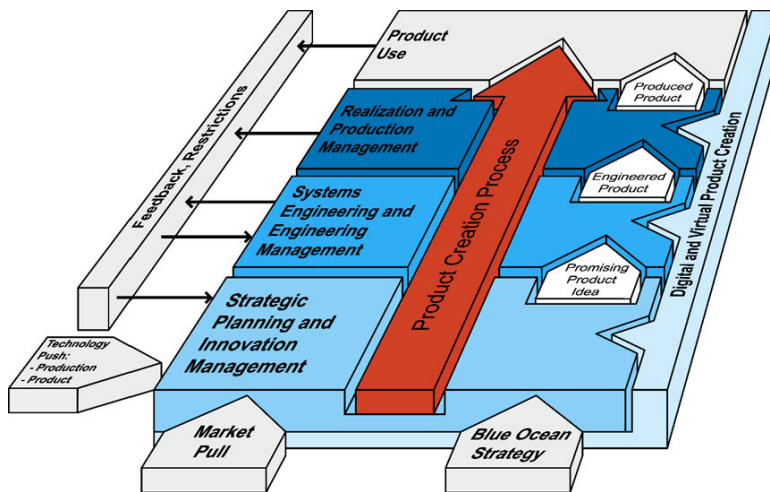


Figure 2: Action field of Product Creation [19].

boundary conditions. Taking these basics into account, the new V-Model was conceived in several iteration steps and further enhancement potentials were continuously elaborated. Step by step the results were validated using the VDI GMA TC 4.10 as a sounding board. Thus, reactions from participants' different angles of experience to the suggested ideas and contents were used as a test of validity. External experts from industry and science enriched the discussions in the TC meetings by presenting their own V-Model-approaches. Potentials described in literature were used as a basis and as an input for the discussions [10, 14, 15, 17, 18]. The final results were presented by the authors and externally validated in a workshop with 25 international experts from science and industry during the International DESIGN 2018 Conference in Dubrovnik.

3 State of the art

In the action field of Product Creation, Systems Engineering and Engineering Management build the core of the Product Creation Process (Figure 2). Out of a promising product idea, a product is engineered [19].

3.1 Mechatronics and cyber-physical systems

In 1969, the Japanese president of YASKAWA Electronic Corporation, Ko Kikuchi, introduced the term “mechatronics” [6, 20]. As manufacturer of automated technical products, such as servo drives and robots, YASKAWA coined the understanding of the term “mechatronics” as the expansion of mechanical components by electronic functions. The term consists of mechanisms and electronics and was protected in the period from 1971 to 1982 as a trade name

[6]. Mechatronic systems are characterized by the functional and spatial integration of sensors, actuators, information processing and a basic system [21]. The functional gain of mechatronic systems compared to electromechanical systems relies on the synergetic effect of interdisciplinary technologies.

Early mechatronic systems of the 1960ies only were constituted by mechanics and electronics. They were not programmable [6], (Figure 3). Step by step, mechatronic systems additionally gained software and became programmable. Further, adaptronics such as anti-lock braking systems emerged.

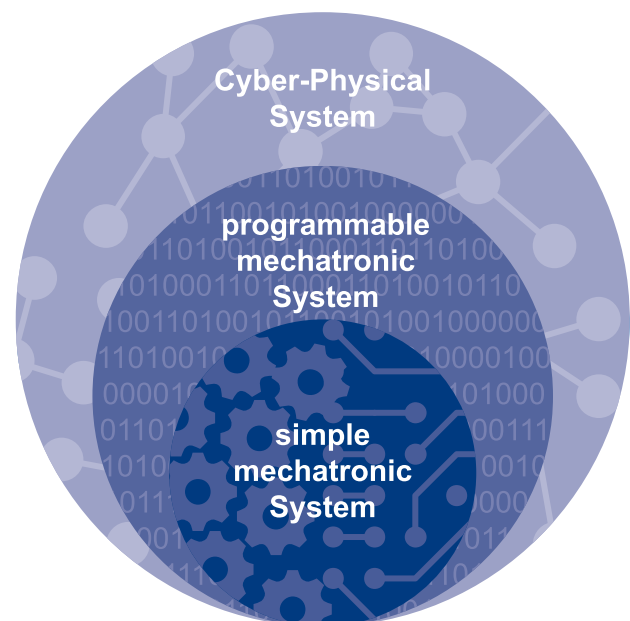
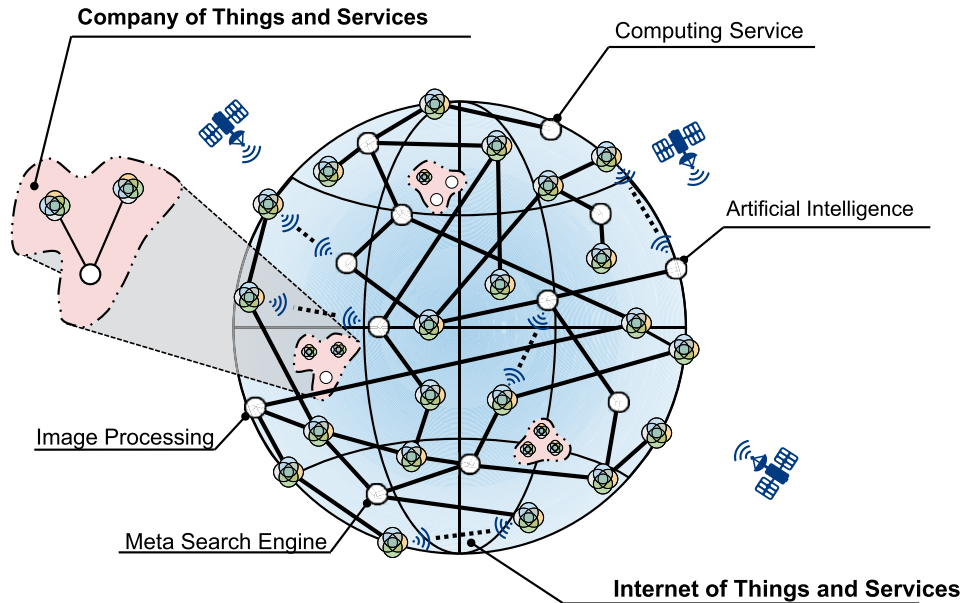


Figure 3: From simple mechatronic systems to Cyber-Physical Systems based on [41].



Legende/ Key

- Data and Services
- ⊗ mechatronic system
- wireless
- wired
- ☁ Company of Things and Services
- 🌐 Internet of Things and Services

Figure 4: Definition of a Cyber-Physical System.

Today, cyber-physical systems (CPS) are created by expanding mechatronic systems with additional inputs and outputs and coupling them to the IoT (Figure 4). With the help of AI, the system properties change during operation. Thus CPS represent the next generation of integrating software into technical systems towards fully software-integrated and networked systems. As illustrated in Figure 4, CPS are interconnected mechatronic systems that are additionally connected to the Internet of Things and Services and adapting their properties during operation. This enables them to communicate with each other and use Internet services. Lee defines CPS as integrations of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa [22]. Broy and Acatech define CPS as close connection between embedded systems for monitoring and controlling physical processes by means of sensors and actuators via communication devices with the global digital networks (the “cyberspace”). This type of system enables a connection between processes of physical reality and the digital network infrastructures available today via chains of effects. This allows diverse applications with high economic potential

and strong innovative strength [23, 24]. For example, the use of CPS in production as Cyber-Physical Production Systems (CPPS) will lead to a fundamental change in production planning and control, expressed by the term “Industry 4.0” [25]. The advantage of CPPS compared to conventional production systems lies in the independent configuration of the production systems for continuous adjustment to the order situation and to changing environmental conditions.

3.2 The VDI 2206 guideline’s V-Model

The original idea of a V-Model for engineering processes was created 1995 by Bröhl and Dröschel in the application field of Software Development [1]. In the core, the symbol of the “V” represents decomposition of the system into its elements on the left thigh and gradual integration of elements and sub-systems into the entire technical system on the right thigh [8]. Besides these two thighs of the “V”, the properties of the product in development are continuously validated and verified [8]. Thus, it is ensured that the “right” system (validation) is developed in the “right” manner (verification) [8]. In the 2004 version, requirements are graphically represented as an input

box. At the bottom of the V-Model, system implementation is represented. Engineering of mechatronic and Cyber-Physical Systems involves multiple disciplines, e. g., mechanics, electrics/ electronics, software as well as pneumatics, hydraulics, optics or others. The result or output of the V-Model is a product. The model is graphically framed by a bracket symbolizing modeling and model analysis, which begins and ends at the middle height of the two thighs. Arrows between the two thighs of the V-Model illustrate the assurance of properties comprising verification and validation activities. At this juncture, the assurance of properties is indicated right-to-left, which gives the impression of a retrospective process [15, 17, 26].

3.3 Systems engineering

Systems Engineering is a structured, multi-disciplinary engineering approach for technical systems, targeting at a cross-disciplinary optimum within a given time frame and budget. For this purpose, the disciplines are structured and networked with each other using models [19]. Besides cross-disciplinary optimum, often mentioned core aspects of Systems Engineering are safety and reliability, complexity management, user experience, lead-time reduction and cost saving. Emphasis is particularly drawn on cross-disciplinarity and frontloading, which describes focused management attention and high capacities spend in early stages of the development process. Systems Engineering comprises technical, organizational and managerial tasks and balances them among each other [10]. Systems Engineering provides a holistic approach, which deals with all stages of life cycle of the System of Interest (SOI) [27, 28]. Restrictions of such upstream and downstream tasks shall be represented in the new V-Model. It is based on Systems Thinking, which includes the clear definition of the system, its elements, its interconnections and the separation from the environment by definition of boundaries [4, 5, 27–31].

4 Need for revision

In addition to the V-Model of the guideline VDI 2206:2004, there is a large number of V-Models, which have a structural similarity to the VDI guideline. These V-Model variants were adapted to individual applications and specific domains [15]. Out of these, representative V-Model variants from Product- and Systems Engineering were chosen and compared with each other in detail [15]. These V-Model

variants differ in their granularity, their application purpose and the envisioned target group [15]. While some represent the entire life cycle in the illustration, others only focus on the phase of product development [15]. An overview of the compared characteristic properties is given in [15].

In addition to the analysis of V-Model variants, literature on the description of product engineering, interdisciplinary work, mechatronics and Systems Engineering was analyzed. In the publications [13, 14, 17] and [8] the resulting potentials are explained in detail. These revision potentials were used as the basis for the discussion in the TC “Interdisciplinary Product Creation”. Three exemplary potentials are provided in the following: First, an integration of the contemporary V-Model into its temporal and contextual environment is missing. The idea of starting with requirements and finishing with a product is too limited. Life-cycle-orientation with the inclusion of all life cycle phases before, in parallel and after development plays a major role in development. This is underlined by the approaches and methods of Systems Engineering, which describe holistic system thinking as a key feature of successful product engineering [32]. Second, requirements elicitation is not integrated into the contemporary V-Model. Due to the illustration as an inbox, the contemporary V-Model gives the impression that requirements were a delimited input. Instead, elicitation, structuring and analysis of requirements are integral tasks of product engineering. Further, requirements collection, modification and management has to be carried out continuously. Since requirements engineering is one of the main success factors of development projects [15], the work with requirements has to be emphasized and the impression of a given input must be prevented. The third potential concerns modeling and model analysis. Models are central in modern product engineering at any point of project maturity. Already in requirements elicitation, models are used to allow the description of connections and structure. The representation and explanation of modeling and analysis in the contemporary V-Model therefore should be universal so that the importance of modeling in development is correctly understood.

5 Validation workshop

Within the validation workshop held at the 15th International DESIGN Conference on May 21st, 2018 at Dubrovnik, Croatia, the current state of work was discussed with 25 international participants from industry and science. The aim of the workshop was to validate the pre-final state of

the V-Model and the guideline VDI 2206:2004. Validation of the described suggestions was structured and supported by elaborated discussion theses. Thus the basis for discovering additional improvement potentials and implications for the revision of VDI Guideline 2206 was provided. Further aims of the workshop were to discuss the integration of examples, the definition of terms and the graphical illustration of the V-Model, in order to include them into the finalization of the guideline.

5.1 Facilitation method

For the validation workshop, the facilitation method “Open Space” was chosen. According to the so-called “coffee-break” anecdote, Harrison Owen invented the open space method when he conducted a congress for 250 organizational developers in 1983. At the end of the conference, participants concluded that the “really useful part” of the otherwise successful meeting had been during the coffee breaks. Owen made this the principle of next congresses. This implies that participants decide “by their feet” which topics they want to discuss out of a selection of given topics. Further, they can even add their own topics. Thus from the participants’ viewpoint, the most important topics are covered. In this case, fifteen prepared discussion theses were offered to the participants. This wide choice was highly appreciated by the participants. They even added two further self-defined discussion theses. In sum, eight theses were selected for deeper discussion. The theses address the main revisions planned to be included in the new V-Model. This includes additions, such as the definition of a Main Feature List and the implementation of checkpoints into the model. The ongoing consideration of requirements management or modeling of model-based development in the entire V-Model are two additional important suggestions.

5.2 Results

The workshop results comprise implications for the revision of the V-Model as well as engineering design research in general. In the following, accepted improvement potentials and further implications for the revision of VDI Guideline 2206 are outlined. Goals and intentions of V-Model revision were generally supported by the participants. On the whole, a broad consensus upon the proposed pre-final revision state of the V-Model and its key elements was achieved. Two further requirements were stressed by

the participants: the inclusion of the engineer as a human being and a comprehensive modeling and analysis. First, the wish to include the human beings with their skills, competencies, convictions and emotions was discussed and taken up in the workshop. In the final version it is represented by coupling the V-Model with the Holistic Product Lifecycle (HPLC) Model, which has been developed by the Technical Committee VDI 702 “VDI System House” and will be published soon. Within the HPLC Model, human beings are represented besides organizational approaches, information and communication technologies (ICT). Thus engineers are provided with a clear orientation and enabled to act as drivers in the design of Digital Business Models. Second, comprehensive modeling and analysis along the whole V-Model was required by the participants. In contrast to the contemporary V-Model of VDI 2206, participants claim a model-based approach of all engineering tasks. As a consequence, the New V-Model is completely framed by the strand “modelling and analysis” (outer blue strand in Figure 5).

Another key finding of the discussion was outlining the necessity of representing requirements engineering within the “V”. This reflects another weak point of the former V-Model version of 2004, where requirements are illustrated as limited input box to upper left thigh of the “V”. Neither the process of requirements elicitation nor requirements management are represented in the contemporary V-Model. This visualization is based on the idealized assumption that requirements could be collected once and completely. In engineering practice however, requirements and their values change along the product development process. As a consequence, requirements elicitation and management is illustrated by a separate strand (inner yellow strand in Figure 5).

Representation of verification, validation, planning verification and validation were another key element of the discussion. According to several Systems Engineering approaches, a set of three arrows is exemplarily representing continuous planning, verification and validation. It was discussed that of course, the system is verified and validated several times along the “V”. In order to simplify the illustration and achieve good clarity, the arrows are included only once and exemplarily. As the system is verified at the same system level, this is represented with a horizontal arrow, while validation is carried out against the next higher level of stakeholder demands and interests and represented by an upwards reaching arrow. Verification and validation are practiced from the very beginning.

The inherent concern logic of the V-Model was consensus among all participants. To prevent the potential misunderstanding of the former V-Model as a process model or

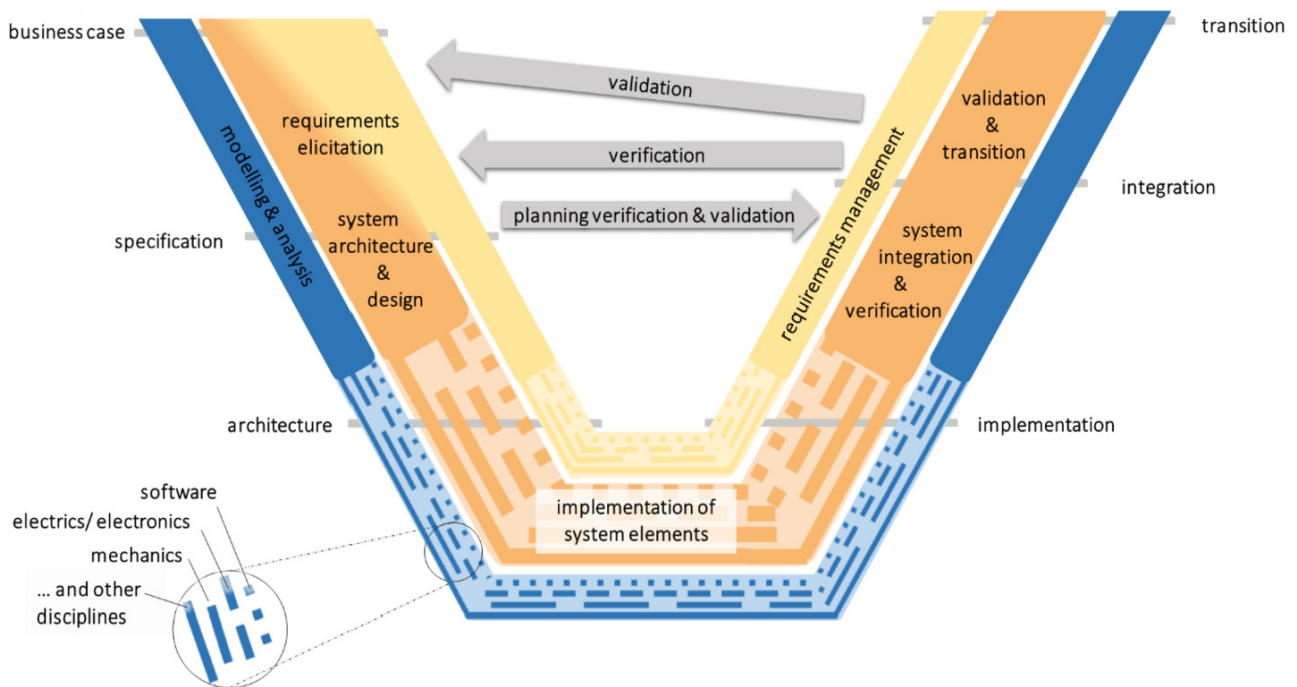


Figure 5: The New V-Model as presented at DESIGN 2018 Conference in Dubrovnik (illustrated by the organization team of the “VALIDATE THE V-MODEL FOR NEW VDI 2206” Workshop Iris Graessler and Julian Hentze).

a time sequence of activities, the graphical layout of the revised version represents the logical sequence of tasks. The key advantage lies in staying independent from the chosen form of project organization. This way the V-Model can be applied in classical project management as well as in an engineering project run by agile principles.

The role of checkpoints was also emphasized by the workshop participants. Nevertheless, one participant stimulated rethinking the naming in order to prevent an association of temporal dependence. A first idea was to name them “revision points” for a consistency check of tasks.

In order to foster industrial implementation of the V-Model, the representation of the “V” shall be self-explaining. Therefore, nestings are not incorporated in the basic illustration of the V-Model. Nestings of V-Models appear on different system hierarchy levels, parallel system elements and subsystems as well as on different maturity levels. Those nestings shall be illustrated using product examples. Such examples are not meant to validate, but to explain tailoring and application of the V-Model.

In addition to the findings concerning the revision of the V-Model, the discussion of theses in the open space revealed further research fields for future engineering methods. These include planning verification and validation aspects, the discussion of milestones as steering elements

in product engineering processes and the implementation of holistic modeling aspects. The results from this workshop can also be seen as an initial set of requirements for a broad variety of methodological approaches towards product engineering. These were gathered not only from an academic perspective, but also from the practical viewpoint of practitioners from companies such as Jaguar Land Rover, Saab, BMW and BST eltomat. These represent a broad portfolio of products in various domains.

6 The new V-Model explanation

Figure 5 illustrates the resulting new V-Model. The V-Model is based on the basic form of the contemporary VDI 2206:2004 Guideline.

6.1 Structure

The new V-model basically consists of three strands. The central strand in orange describes the core activities and tasks. The inner, yellow strand describes the handling and the work with requirements. The outer, blue strand represents the modeling and analysis activities. Each of the three strands graphically contains the involved disciplines

at the bottom of the V-Model. This graphic representation shall symbolize that implementation of system elements relies on deep and interwoven detailing in and between the involved disciplines. This is underlined by a graphical consistent background in the individual strand color, since the exchange and the coordination among the disciplines is a crucial success factor. Possible disciplines are classic mechatronic disciplines such as mechanics, electrics/electronics and software as well as pneumatics, hydraulics, optics, human science, etc.. These disciplines are relevant in the main strand as well as in modeling and analysis and in requirements engineering. A brief description of key elements of the new V-Model will be given below. The detailed description and the related examples are explained in the new guideline published in June 2020.

6.2 Core activities and tasks

Requirements elicitation

Elicitation, structuring and analysis of requirements form the basis of the development process. From needs and wishes of the different stakeholders (stakeholder needs), requirements are derived and documented in technically and formally language and form, e. g., in a requirements list or diagram. The requirements are described from the perspective of the stakeholder and do not describe the system in the first step, but the needs and wishes of the stakeholders (stakeholder requirements). These are subsequently converted to the system view in the following step (system requirements) [4]. The specification, which is the result of the requirements elicitation, is a description of the system to be developed with a usually huge number of requirements. In order to develop a specification that is as complete as possible, the new guideline offers an auxiliary instrument: The Main Feature List for Mechatronic and Cyber-Physical Systems. This Main Feature List supports the user of the guideline to achieve completeness and a high quality of the specification [33]. Essential quality criteria for a product specification are completeness, clarity, consistency and correctness [34]. In the further course of development, the requirements list will be supplemented by successive concretisation and detailing of the product properties. The objective is to have all essential requirements quantitatively defined at the end of system architecture. In addition, requirements can change as a result of feedback and iterations from downstream sections or by updating customer wishes or market requirements. Changes are therefore systematically tracked and evaluated. Result of requirements elicitation is a description of the system characteristics which are to be fulfilled.

The target parameters for property assurance are also derived from the requirements.

System architecture and design

The core of interdisciplinary Product Engineering takes place in the phase of system architecture and design. In the sub-section “system architecture” a cross-disciplinary overall solution structure of the system is developed. Based on the functional structure and the underlying principles of action, the system architecture comprises the building structure in the mechanical sense, signal flow structures and circuit diagrams in the electronic sense, as well as the structuring of a software program into its modules and components, including the respective interfaces, from the perspective of software. The system architecture thus embodies elements, relationships and necessary principles for the design and further development of a system [4]. Structural alternatives are compared on the basis of comprehensible criteria, such as the realisation of interdependencies, functional analyses and the consideration of exclusion criteria. Ideally, this evaluation is carried out with the participation of users. The best possible allocation of the individual functions to the disciplines involved is determined in an iterative procedure on the basis of system partitioning. The task of the system architect is the decomposition of the system into implementable units. Starting from the required system functions and properties, a system is broken down step by step from the functional, property and implementation point of view. The result is verifiable units, i. e., system components with allocated specified requirements and relevant interface information. The number and design of the interfaces within an architecture and to the system boundary significantly influences the simplicity, adaptability and testability of a system.

Implementation of system elements

In the implementation of system elements, the specified system elements are laid out, dimensioned, designed and detailed. Mechanical components are dimensioned using Computer Aided Design (CAD), Finite Element Methods (FEM), calculations and simulations. For example, static and dynamic basic laws are applied in the design, stresses are calculated, deformations and heating under load are analysed and service lives are calculated. This is followed by a detailing of the components up to the determination of the tolerances and fits required for the fulfillment of the function as well as all manufacturing regulations. Mechanical components include housings, covers and flanges of control units in addition to those used

to transmit forces or moments. For electrical and electronic components, for example, associated printed circuit boards are designed and the electronic components selected, Application-Specific Integrated Circuits (ASICs) specified or reconfigurable circuits (field programmable gate arrays, FPGAs) selected as prefabricated components. The classic Printed Circuit Board (PCB) is a printed circuit and is used for mechanical fastening and electrical connection of the components. Electronic Design Automation (EDA) or electronic CAD (ECAD) as well as simulations of electronics and electromagnetism (FEM, FDTD, FIT) are used for design automation. In addition to the circuits, sensors and actuators are developed or selected and the electrical infrastructure is designed. For software elements, the implementation includes the conversion of an algorithm or software design into a computer program. Based on the software architecture, functions are implemented using algorithms and the associated code is generated. Database models are implemented by implementing the modelling in concrete schemes. Furthermore, the user or system interface is designed. During detailing and implementation, the respective mechanical, electronic and software interfaces are taken into account. At the same time, this section serves to solve problems that have arisen during system integration.

System integration and verification

The integration step-by-step merges the implemented system elements and subsystems into the next higher product-hierarchy level. The goal is that the integrated overall system meets the formerly specified requirements. In the system integration, mechanics, hardware, software as well as service offers are combined to a system or step by step to system elements of a higher system level. System integration takes place in close interplay with detailing and implementation, so that any problems that arise are resolved immediately. The interaction of the components or elements of the disciplines involved is examined comprehensively in the form of a system simulation or virtual commissioning. The integration strategy must be agreed in advance with all relevant stakeholders.

The property prognosis and assurance is carried out during development. The system properties to be expected are forecasted at an early stage on the basis of models. The property validation that accompanies the development is used for the continuous examination of the pursued solution on the basis of investigations with virtual prototypes (computer simulation), physical prototypes (hardware experiment) or their combination (e. g., hardware-in-the-loop, HiL). The progress of the product is thus checked

against the required system properties in a virtual and/or physical environment. The property validation includes verification and validation. Verification is the confirmation by objective proof that a specified requirement is fulfilled [35, 36]. It answers the question: “Was the system built correctly? Requirements are verified at the same system level at which they were specified. Each requirement must be verified by an appropriate measure, e. g., analysis, inspection, demonstration or test. A verification measure can also demonstrate multiple requirements. It should be performed as early as possible and at the lowest possible system level.

Due to the close interdependence of integration and property assurance, it must be planned in advance in which stages the system will be integrated and for which required scopes with which method, when and at which system level the implementation shall be proven. Basically, the following methods for property assurance are distinguished: (1) Theoretical Analysis: Calculation, Modelling and Simulation; (2) Inspection: for easily visible, measurable and recognizable properties; (3) Demonstration: qualitative proof of functionality, usually with minimal instrumentation; (4) Test: quantitative proof, usually in a defined test environment; The integration into the next higher system level must not start until the verification of the initial level has been completed.

Validation and transition

Individual system elements and subsystems have to be validated; the approach depends on the individual planning of the system or system element. In any case, the fully integrated overall system is validated against the stakeholder needs. At the end of the logical pass through the V-Model, the system, whether it be a physical system, a prototype or a simulation is handed over to another entity or stakeholder. This can be a customer, but also another department, a colleague or a subsequent project phase for iterative development. Validation proves that the work result can be used by the user for the specified application [35, 36]. The control questions are: “Was the right system built? Is the customer satisfied? Were the requirements right? Do they represent the needs of the stakeholders?” Validation is carried out with the customer or between the requirements provider and the recipient. The validation of the system takes place with the superordinate requirements (next higher system level) and serves as proof of the fulfillment of stakeholder needs or customer value. Each requirement must be verified and validated. In the event of fuzzy requirements, suitable validation measures, such

as acceptance tests, are agreed with the stakeholders during the planning phase. In the case of large quantities and safety-critical applications, the product is not only specified by models, but also functionally secured using prototypes. All necessary models and documents for the subsequent stages in the product life cycle are stored and structured using Product Data Management (PDM) along the product development process. From this, the final product documentation is built up and completed. As a prerequisite for a later digital twin [42], basic models are created which allow the simulation of system behaviour. This is also known as the Digital Master. The digital representation of the associated customer-specific versions of the product is referred to as the Digital Twin.

6.3 Additional general activities

Modeling and analysis

For efficient computer-based development of mechatronic and cyber-physical systems, a digital representation of the system is required. Modern product development methods are consistently supported by models. Models have to be created for the overall system, all subsystems and system elements. Dependencies of subsystems and system elements on a high system level are taken into account as well as dependencies into and between the involved disciplines [6]. Even requirements elicitation and analysis methods generate computer-aided models: requirement diagrams or stakeholder models are examples. As a result, a large number of models is generated during the development of system architecture and implementation. As graphically illustrated in the New V-Model, modeling and analysis also affects the individual disciplines: Each involved discipline has a huge variety of models. In context of model-based development and MBSE (Model-based Systems Engineering), the models of the disciplines are linked to the models on system level and have an active exchange of information. Forming the models also requires active analysis and synthesis [37]. This concerns the evaluation, the interpretation and the simulation of system elements, subsystem or the entire system.

Requirements Engineering

In a multitude of definitions, Requirements Management is described in conjunction with Requirements Elicitation as “Requirements Engineering” [38, 39]. In order to stress the importance of elicitation as the basis for a successful project, both terms are used individually in the V-Model.

Requirements management describes the handling of requirements. The structuring, allocation, analysis and integration of changes of requirements has to be handled during the whole development project. Handling requirements is supported by various tools, databases and methods. Requirements management is relevant on high system level as well as on the level of system elements. The different system levels have high influences on each other, so changes in requirements must be reflected and documented to the overall project and all relevant project participants.

6.4 Checkpoints

The three strands that represent the main tasks and activities in the New V-Model are backed by a structure of six checkpoints. The checkpoints contain questions, which help the user of the V-Model to carry out a structured and complete development process leading towards a successful product. The six checkpoints are described in the publication [13]. The questions serve for orientation and have no claim of completeness. They may also encourage additional questions for individual needs or standards on company or individual project level. In contrast to project management, in which decision gates or milestones are described as obligatory control points with the possibility of terminating a project, the checkpoints give hints to the user [40]. Checkpoints do not represent a structure of a project, but deliver a substantive support for application [13]. Two exemplary checkpoints at the specification and for the integration are provided in Tables 1 and 2.

6.5 Assurance of properties

In the middle of the model, between the two thighs of the V, three arrows symbolically indicate the activities of the assurance of properties. The individual arrows are representative of activities that can be highly relevant at almost any time. “Planning verification & validation” describes the need to start extensive activities already on the left thigh of the V-Model in order to be able to secure the defined requirements in the sequence. The planning goes beyond the description of a good requirement. It not only describes how a request needs to be secured, but also how it should be done. The representative arrows “Validation” and “Verification” divide the assurance of properties into the questions “Did we do it right?” (verification) and “Did we do the right thing?” (validation). Validation is illustrated by an upward arrow towards a higher level, because

Table 1: Checkpoint Specification.

Checkpoint 2: Specification
<input type="checkbox"/> Have all relevant requirements been derived from the main list of characteristics? (all secondary characteristics recorded or checked?) <input type="checkbox"/> Are the requirements clearly formulated and/or measurable? <input type="checkbox"/> Are the requirements complete, correct and consistent? <input type="checkbox"/> Are the required specifications in the requirements comprehensible and realistic? <input type="checkbox"/> Is the specification formulated in a solution-neutral way (independent of effect/active principle)? <input type="checkbox"/> Are the specified requirements verifiable and are the methods and means for verification and validation known and available? <input type="checkbox"/> Has a distinction been made between fixed and desired requirements or optimization goals? <input type="checkbox"/> Has a process for changing the requirements been agreed?

Table 2: Checkpoint Integration.

Checkpoint 5: Integration
<input type="checkbox"/> Are the interfaces technically implemented? (bus, message format, mechanical fit etc.) <input type="checkbox"/> Are the "test/measurement methods" and framework conditions known and feasible? <input type="checkbox"/> Are sufficient integration instructions (sequence etc.) available? System element → Sub-system → System <input type="checkbox"/> Were given tolerances, test sequences and standards observed? (CE, ISO etc.) <input type="checkbox"/> Are there any deviations from the specification and have necessary changes been communicated? <input type="checkbox"/> Have potential errors been identified and are their effects known? <input type="checkbox"/> Has testing been consistently performed at all levels of the system hierarchy? <input type="checkbox"/> Have aspects of preventive quality management been taken into account in the assurance of properties? (e.g. FMEA) <input type="checkbox"/> Does the product meet the current legal requirements?

not the technical specification is checked, but the fulfillment of the wishes of the customer and other stakeholders.

7 Summary and outlook

The high level of agreement with the discussed theses validates the V-Model elaborations of the workshop organizers. None of the proposed changes was rejected. All discussed changes were confirmed and their importance was stressed. Additional suggestions on formulation and presentation were reflected in the final version of the guideline. The dissemination of examples and their presentation in the guideline raise awareness and applicability. During the workshop, some new members could be recruited to join the TC 4.10 and from then on actively contributed to the elaboration and formulation of the guideline. In June 2020, the New V-Model was published as a "VDI Green Print".

Acknowledgment: We would like to thank the members of the technical committee VDI GMA 4.10 "Interdisciplinary Product Creation" and all workshop participants at the DESIGN 18 conference for the lively technical discussions.

References

1. United States Department of Defense, "Systems Engineering Fundamentals," Fort Belvoir, Virginia, Jan. 2001.
2. U.S. Department of Transportation, "Systems Engineering Guidebook for Intelligent Transportation Systems: Version 3.0," 2009.
3. NASA, *NASA Systems Engineering Handbook*. Washington D.C.: United States Government Printing Office, 2007.
4. D. D. Walden, G. J. Roedler, K. Forsberg, R. D. Hamelin and T. M. Shortell, *Systems engineering handbook: A guide for system life cycle processes and activities*, 4th ed. Wiley, 2015.
5. I. Gräßler, J. Hentze and C. Oleff, "Systems Engineering Competencies in Academic Education: An industrial survey about skills in Systems Engineering," in *13th System of Systems Engineering Conference*, Paris, 2018, pp. 542–547.
6. VDI 2206 – Design methodology for mechatronic systems, 2206, Beuth Verlag, 2004.
7. A.-P. Bröhl (Ed.), *Das V-Modell: Der Standard für die Softwareentwicklung mit Praxisleitfaden*, 2nd ed. München: Oldenbourg, 1995.

8. I. Gräßler, "A new V-Model for interdisciplinary product engineering," (eng), in *Engineering for a changing world: 59th IWK, Ilmenau Scientific Colloquium*, Technische Universität Ilmenau, September 11–15, 2017: proceedings, 2017.
9. I. Gräßler, "Umsetzungsorientierte Synthese mechatronischer Referenzmodelle: Implementation-oriented synthesis of mechatronic reference models," in *Konferenzband der VDI Mechatronik: Fachtagung Mechatronik 2015*, T. Bertram (Ed.), Dortmund, 2015, pp. 167–172.
10. I. Gräßler and J. Hentze, "Enriching Mechatronic V-Model by Aspects of Systems Engineering," in *2015 – Smart Structures and Materials*, A. L. Araujo and C. A. Mota Soares (Eds.), 2015, pp. 80–86.
11. I. Gräßler, A. Pöhler and J. Hentze, "Decoupling of product and production development in flexible production environments," in *27th CIRP Design Conference 2017*, Cranfield Campus, Cranfield, United Kingdom, 2017, pp. 548–553.
12. J. Hentze, T. Kaul, I. Gräßler and W. Sextro, "Integrated modeling of behavior and reliability in system development," in *ICED17: 21st International Conference on Engineering Design Vancouver*, 2017, pp. 385–394.
13. I. Gräßler, "Competitive Engineering in the Age of Industry 4.0 and Beyond," in *Proceedings of TMCE*, Las Palmas de Gran Canaria, 2018.
14. I. Gräßler and J. Hentze, "A V-model based comparison of Systems Engineering approaches," in *ECEC 2015*, Lisbon, Portugal, 2015, pp. 80–88.
15. I. Gräßler, J. Hentze and T. Bruckmann, "V-Models for Interdisciplinary Systems Engineering," in *Proceedings of the DESIGN 2018 15th International Design Conference*, Dubrovnik, 2018, pp. 747–756.
16. I. Gräßler, C. Oleff and J. Hentze, "Role Model for Systems Engineering Application," in *22nd International Conference on Engineering Design*, Delft, The Netherlands, 2019.
17. I. Gräßler, J. Hentze and X. Yang, "Eleven Potentials for Mechatronic V-Model," in *Production Engineering and Management: 6th International Conference*, Lemgo, Germany, 2016, pp. 257–268.
18. G. Versteegen, *Das V-Modell in der Praxis: Grundlagen, Erfahrungen, Werkzeuge*. Heidelberg: dpunkt.verlag, op. 2000.
19. I. Gräßler, "Implementation-oriented synthesis of mechatronic reference models," in *Fachtagung Mechatronik 2015*, Dortmund, Deutschland, 2015, pp. 167–172.
20. F. Harashima, M. Tomizuka and T. Fukuda, "Mechatronics – "What Is It, Why, and How?": An editorial," *IEEE/ASME Trans. Mechatron.*, vol. 1, no. 1, pp. 1–4, 1996.
21. I. Gräßler, *Kundenindividuelle Massenproduktion: Entwicklung, Vorbereitung der Herstellung, Veränderungsmanagement*. Springer Berlin, Heidelberg, New York u. a., 2004.
22. E. A. Lee, "Cyber Physical Systems: Design Challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, Orlando, FL, USA, 2008, pp. 363–369.
23. Acatech, *Cyber-Physical Systems: Innovationsmotoren für Mobilität, Gesundheit, Energie und Produktion*. Dordrecht: Springer, 2011.
24. M. Broy, *Cyber-Physical Systems: Innovation Durch Software-Intensive Eingebettete Systeme*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2010.
25. H. Kagermann, W.-D. Lukas and W. Wahlster, "Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution," in vol. 13, *VDI-Nachrichten*, 2011.
26. J. Gausemeier and S. Moehring, "New Guideline 2206: A flexible procedure model for the design of mechatronic systems," in *Proceedings of ICED 03, the 14th International Conference on Engineering Design*, Stockholm, 2003.
27. IEEE, *Systems and software engineering: System life cycle processes*, 2nd ed. Geneva: ISO/IEC-IEEE, 2008.
28. R. Haberfellner, *Systems Engineering: Grundlagen und Anwendung*, 12th ed. Zürich: Orell Füssli, 2012.
29. C. W. Churchman, *The systems approach*, 2nd ed. New York, NY: Laurel Book, 1984.
30. L. van Bertalanffy, *General system theory: Foundations, development, applications*, 14th ed. New York: Braziller, 2003.
31. R. D. Arnold and J. P. Wade, "A Definition of Systems Thinking: A Systems Approach," *Procedia Computer Science*, vol. 44, pp. 669–678, 2015.
32. E. C. Honour, "Understanding the Value of Systems Engineering," 2014. Willey. DOI:10.1002/j.2334-5837.2004.tb00567.x.
33. I. Gräßler, M. Dattner and M. Bothen, "Main Feature List as core success criteria of organizing Requirements Elicitation," Milan, 2018. DOI:10.17619/UNIPB/1-679.
34. I. Gräßler, V. Haas and W. Suchowskyj, "Innovation Based on Applying Design Methodology," in *TMCE 2012*, 2012, pp. 37–43.
35. ISO 29119, *Software and systems engineering: Software testing: Part 1: Concepts and definitions = Ingénierie du logiciel et des systèmes: essais du logiciel. Partie 1, Concepts et définitions*. Geneva, s. l., New York: ISO, 2013.
36. M. Daigl and R. Glunz, *ISO 29119 – Die Softwaretest-Normen verstehen und anwenden*. Heidelberg: dpunkt.verlag, 2016.
37. K. Pohl, H. Hönninger, R. Achatz and M. Broy, *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
38. C. Ebert, *Systematisches Requirements Engineering: Anforderungen ermitteln, spezifizieren, analysieren und verwalten*, 4th ed. Heidelberg: dpunkt.verlag, 2012.
39. K. Pohl and C. Rupp, *Requirements engineering fundamentals*, 2nd ed. Santa Barbara, CA: Rocky Nook, 2015.
40. R. G. Cooper, "Perspective: The Stage-Gate® Idea-to-Launch Process—Update, What's New, and NexGen Systems," *J Product Innovation Man*, vol. 25, no. 3, pp. 213–232, 2008.
41. M. Abramovici and O. Herzog (Eds.), "Engineering im Umfeld von Industrie 4.0", 2016.
42. R. Stark and T. Damerau, "Digital Twin", in *International Academy for Production Engineering*, 2019.

Bionotes

Iris Graessler

Chair for Product Creation at the Heinz Nixdorf Institute, Paderborn University, Germany

iris.graessler@hni.upb.de

Prof. Dr.-Ing. Iris Gräßler studied mechanical engineering at the RWTH Aachen University, where she received her doctorate degree in 1999. In 2003 she qualified as a professor at the RWTH Aachen University due to her habilitation thesis entitled “Development of Configurable Adaptive Mechatronic Systems”. After thirteen years as a manager at Robert Bosch GmbH in the fields of product development, production systems and change management, she has held the Chair for Product Creation at the Heinz Nixdorf Institute of the University of Paderborn since 2013. Her research focuses on model-based product development, especially agile scenario technique, requirements management, systems engineering, development methodology, digitalization and virtualization. The emphasis lies on adaptive configurable mechatronic systems and cyber-physical systems. In order to further develop and transregionally harmonize development methods, Graessler commits herself to the VDI (Association of German Engineers) work on guidelines such as the reissue of the directive VDI 2206 in 2020. As a result of her scientific achievements, she was appointed to the Scientific Society for Product Development (WiGeP) in 2017 and to the Advisory Board of the Design Society in 2019. Further functions include the Scientific Advisory Board of the VDI/VDE Society for Measurement and Automation Technology (GMA), the Scientific Advisory Board for the Application Center Industry 4.0 Potsdam, Chairwoman of the Examination Committee Industrial Engineering, Board of Directors of the Heinz Nixdorf Institute and the Scientific and Technological Advisory Board of the Linz Center of Mechatronics GmbH.

Julian Hentze

Dr. August Oetker Nahrungsmittel KG Bielefeld, Industry 4.0

Strategy & Competence, Bielefeld, Germany

julian.hentze@oetker.com

Julian Hentze studied mechanical engineering at Paderborn University and had subsequently worked at the chemical company Flint-Group in Ahaus. In 2014 he returned to Paderborn to join the Chair for Product Creation of Prof. Dr.-Ing. Iris Gräßler at the Heinz Nixdorf Institute. Since January 2020 Julian Hentze has been a Senior Technology Engineer at Dr. August Oetker Nahrungsmitteltechnik KG in Bielefeld.