J. of Design Research

ISSN online: 1569-1551 - ISSN print: 1748-3050
https://www.inderscience.com/jdr

Hybrid development of physical products based on systems engineering and design thinking: towards a new process model

Frank Koppenhagen, Tim Blümel, Tobias Held, Christoph H. Wecht, Paul D. Kollmer

DOI: 10.1504/JDR.2024.10068261

Article History:

Received:	29 January 2024
Last revised:	20 October 2024
Accepted:	21 October 2024
Published online:	03 January 2025

Hybrid development of physical products based on systems engineering and design thinking: towards a new process model

Frank Koppenhagen*

Department of Mechanical Engineering and Production Management,
Hamburg University of Applied Sciences (HAW Hamburg),
Hamburg, 20099, Germany
Email: frank.koppenhagen@haw-hamburg.de
*Corresponding author

Tim Blümel

Exterior Development of Aerodynamic Components,
Porsche AG,
Weissach, 71287, Germany
Email: Tim.bluemel@t-online.de

Tobias Held

Department of Mechanical Engineering and Production Management,
Hamburg University of Applied Sciences (HAW Hamburg),
Hamburg, 20099, Germany
Email: tobias.held@haw-hamburg.de

Christoph H. Wecht

Faculty of Engineering and Business,
New Design University,
St. Pölten, 3141, Austria
Email: christoph.wecht@ndu.ac.at

Paul D. Kollmer

Institute of English and American Studies,
University of Hamburg,
Hamburg, 20146, Germany
Email: paul.kollmer@uni-hamburg.de

Abstract: Hybrid process models combine elements of agile approaches with stage-gate processes from systems engineering to address the specific challenges of developing physical products. However, there is currently no established hybrid product development process model that has been widely adopted in the industrial practice of new product development (NPD). The aim of this paper is to develop a hybrid product development process model that has a high practical suitability for the development of physical products. Our paper begins by defining the fundamental principle of agile development with regard to engineering design methodology. Subsequently, we analyse Stanford's hybrid ME310 process model's practical suitability for the development of physical products, both theoretically and empirically. Based on the process model's limitations identified in these analyses, we create a new hybrid process model called 'Systematic Engineering-Design-Thinking' (SEDT), which builds on the ME310 process model but integrates essential methods from systems engineering to improve solution space exploration.

Keywords: agile product development; hybrid product development; design thinking; systems engineering; product design; product innovation; product development process models; design theory and methodology.

Reference to this paper should be made as follows: Koppenhagen, F., Blümel, T., Held, T., Wecht, C.H. and Kollmer, P.D. (2024) 'Hybrid development of physical products based on systems engineering and design thinking: towards a new process model', *J. Design Research*, Vol. 21, Nos. 3/4, pp.210–261.

Biographical notes: Frank Koppenhagen is a Professor of Engineering Design and Product Development at the Hamburg University of Applied Sciences in Hamburg, Germany and has held this position since 2009. He received his PhD in Mechanical Engineering from the Technical University of Hamburg in 2004 and worked in different capacities in passenger car development at the Mercedes-Benz Technology Center in Sindelfingen, Germany before he was appointed Professor. His research focuses on engineering design methodology and modular product architectures. He has authored and co-authored several publications in these fields. In 2012, he received the Hamburg Award for excellence in academic teaching. In 2019, he was Visiting Professor at the Center for Design Research at Stanford University in Palo Alto, CA, where he researched hybrid product development approaches based on design thinking. In 2024, he was Visiting Professor at the Design Engineering Lab of Oregon State University, where he researched technology trajectories and the emergence of dominant designs.

Tim Blümel received his BSc in Mechanical Engineering with a major in Engineering Design and Product Development from the Hamburg University of Applied Sciences, Hamburg, Germany, in 2020, and the MSc in Technology Management from the University of Stuttgart, Stuttgart, Germany, in 2023. Since 2023, he has been working as a Design Engineer in the field of exterior development of aerodynamic components with Porsche AG, Weissach, Germany.

Tobias Held graduated from the University of Hamburg as Dipl.-Ing. oec. and received his PhD in Industrial Engineering from the Technical University of Hamburg in 2002. From 2003 to 2006 he worked as an innovation and logistics planner at Unilever. Since 2006 he has been a Professor, first at the West Coast University of Applied Sciences, Heide, Germany, and since 2009 at the Hamburg University of Applied Sciences, Germany. He has authored and co-authored more than 50 scientific publications.

Christoph H. Wecht is a Professor of Management at the New Design University (NDU) in St. Pölten, where he leads the Bachelor's program in Management by Design. He publishes scholarly and practical journal papers and book contributions, and is also active as a coach and lecturer. In the summer of 2019, he was a Guest Professor at the Center for Design Research (CDR) with Prof. Larry Leifer, PhD at Stanford University in Palo Alto, USA, where he deepened his studies in Design Thinking and Customer Centricity. In addition to his professorship at the New Design University, he holds a teaching assignment for technology management at the University of St. Gallen (HSG) and teaches as a lecturer at their Executive School (ES-HSG).

Paul D. Kollmer is currently a scientific employee at the University of Hamburg's Institute of English and American Studies and has held this position since 2019. Previously, he received his Master's degree in Literary and Cultural Studies within the same institute. From 2013 to 2015, he was a grantee through the Fulbright United States Teaching Assistant program. He is also a 2013 graduate of New York University's College of Arts and Science where he received a Bachelor of Arts degree in German, Comparative Literature and Sociology.

This paper is a revised and expanded version of a paper entitled 'The challenging combination of agility and convergence in hybrid product development processes: an empirical analysis of Stanford's ME310 process model' presented at the *23rd International Conference on Engineering Design (ICED21)*, Gothenburg, Sweden, 16–20 September, 2021.

1 Introduction

1.1 Initial situation and problem statement

In recent decades, manufacturing companies have had to face the challenges of shorter technology and product life cycles, increased customer requirements, and greater product complexity (Ehrlenspiel et al., 2020; Ulrich et al., 2020; Schuh and Dölle, 2021). A high degree of flexibility and adaptivity has become a crucial competitive factor, as are innovative capability and customer orientation in new product development (NPD) (Soltani et al., 2013; Kettunen et al., 2015). Although the traditional stage-gate process from systems engineering shows considerable limitations in dealing with these competitive factors (Sommer et al., 2015), it is "*still the most widely adopted NPD process*" (Cocchi et al., 2021). The separation of problem analysis and solution development typical of such stage-gate processes is well suited for systematic solution development of well-defined problems, such as the development of a successor product generation. However, since radical product innovations usually require a fundamentally new understanding of the problem, which is often only gained during solution development, stage-gate processes tend to create rather incremental innovations (Bagno et al., 2017). Separating the understanding of the problem from solution development also makes it difficult to adapt flexibly to a dynamically changing environment. The so-called agile development approaches, which originated in software development, have therefore recently gained attention in the development of physical products (Atzberger et al., 2020; Conforto et al., 2014; Schmidt et al., 2018). However, the understanding of agile development differs significantly. It can refer both to a mindset and culture ("*being agile*") and to the use of specific agile methods and practices ("*doing agile*") (Denning, 2016; Goevert et al., 2019). Thus far, a consistent definition of the fundamental principle of agile development of physical products in terms of engineering design methodology is lacking. This paper's understanding of physical products are mechatronic products that not only have a signal flow in their electronic components, but also an energy and/or material flow in their mechanical carrier system.

Attempts to directly transfer agile development approaches known from software development to the development of physical products have not yielded the desired success (Simpson and Hinkle, 2018). In practical implementation, a variety of difficulties emerged, the central causes of which are rooted in the so-called '*constraints of physicality*' (Ovesen, 2012; Schmidt et al., 2018; Schmidt et al., 2017), which can essentially be traced back to two fundamental problems:

1 The effort and time required for building and testing physical prototypes, especially when involving potential customers (Schmidt, 2019; Ovesen, 2012; Cooper and Sommer, 2018) as well as

2 The process-related and contractual integration of suppliers in the course of increasingly inter-company collaborative product development. (Schröder, 2020; Klein, 2016; Atzberger et al., 2020).

As a consequence, hybrid product development process models combining different methods of agile product development with stage-gate processes from systems engineering have been developed (Cooper and Sommer, 2016a, 2016b; Cooper and Sommer, 2018; Edwards et al., 2019; Ovesen and Sommer, 2015; Heimicke et al., 2020). Said approaches differ both in terms of the selection and scope of agile methods and in terms of the development phases into which these methods are integrated (for a current systematic literature review of hybrid product development processes see Cocchi et al. (2024)). However, there is currently no established hybrid process model for the systematic development of physical products.

1.2 Aim and scope of this paper

The knowledge gaps identified in Section 1.1 give rise to two central research questions of high theoretical and practical relevance:

1 What is the fundamental principle of agile and, derived from this, hybrid development of physical products in terms of engineering design methodology?

2 What might a feasible hybrid process model for the development of physical products look like?

Among the currently existing agile product development approaches, design thinking in particular has the potential to develop a fundamentally new understanding of an existing problem ("reframing") due to its pronounced user-centricity (Cocchi et al., 2021). For this reason, the agile part of the hybrid product development process model to be newly developed should be based on a design thinking approach. The starting point for developing the new process model is the ME310 process model developed at the Center for Design Research at Stanford University. Rooted in a design thinking approach, this process model also defines milestones to structure the entire product development process and to control the convergence of solution development. The ME310 process model has been used for several years in Stanford University's ME310 Design Innovation Course to develop innovative products for real-world design challenges from cooperating industrial companies. We analyse the ME310 process model´s practical suitability for developing physical products both theoretically and empirically. From the process model´s limitations identified through these analyses, we derive specifications for the design of a new hybrid process model. Finally, we create a new hybrid process model called 'Systematic Engineering-Design-Thinking (SEDT)'. This process model builds on the ME310 process model's macro-logic but integrates essential methods from system engineering to improve solution space exploration and convergence of the development process.

1.3 Structure of the paper

In Section 2, we first define the fundamental principle of agile product development in terms of engineering design methodology and illustrate its implications for the development of physical products from which the necessity for a hybrid development process can be derived. We then delineate the different semantic and methodological taxonomy levels of Design Thinking in the context of product development and introduce the ME310 process model. In Section 3, we investigate the practical suitability of the ME310 process model for the development of physical products through theoretical and empirical analysis, and derive specifications for the development of a new hybrid process model. The newly developed process model is presented in Section 4. Section 5 provides a conclusion and outlines the need for further research.

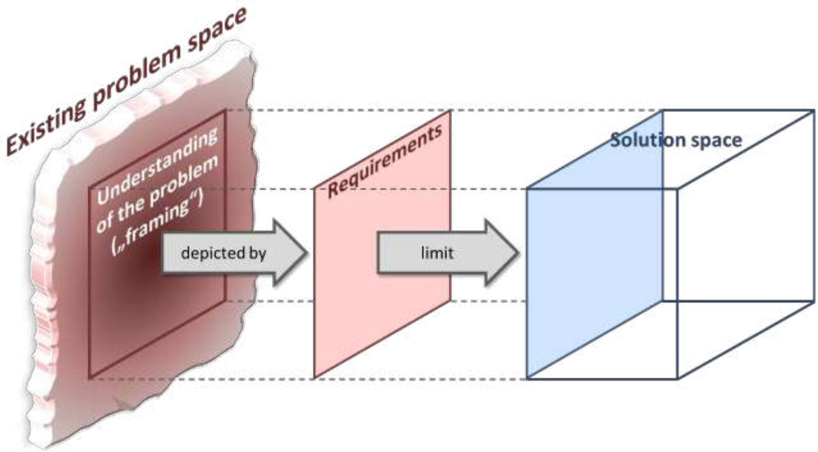
2 Theoretical foundation

2.1 The fundamental principle of agile product development

> "The situation is complex and uncertain, and there is a problem in finding the problem." (Schön, 1983)

Product development processes are creative problem-solving processes in which the underlying problem often cannot be clearly and conclusively defined. Thus, the problem space, primarily involving cognitive understanding, as well as the solution space, focusing on technical possibilities, are generally open. The development of problem space and solution space are mutually dependent since the understanding of the problem depends on conceivable solutions to it (Rittel and Webber, 1973). The development of an understanding of the problem and its representation by formulating requirements on different aggregation levels is therefore an initial creative act, which limits the solution space and already contains a preliminary vision of the solution. This process is referred to as the problem formulation (Figure 1).

Figure 1 Understanding of the problem, requirements and solution space (see online version for colours)



Source: cf. Ponn and Lindemann (2011)

A product development process can be described as a transformation process. Starting from an actual user's need, an understanding of the problem must first be acquired and then operationalised by formulating requirements before the actual solution can be developed, which, in turn, is finally intended to satisfy the actual user's need (Figure 2). At each step of this transformation process, deviations and information loss can occur, which may lead to a misfit between the development result and the user's need. Agile product development methods, such as design thinking, try to prevent such mismatches through a concomitant iterative development of problem understanding and solution. This co-evolution of problem and solution space (Dorst and Cross, 2001) is guided by two control loops: a validation and a verification control loop (Figure 3).

Figure 2 The product development process as transformation process (see online version for colours)

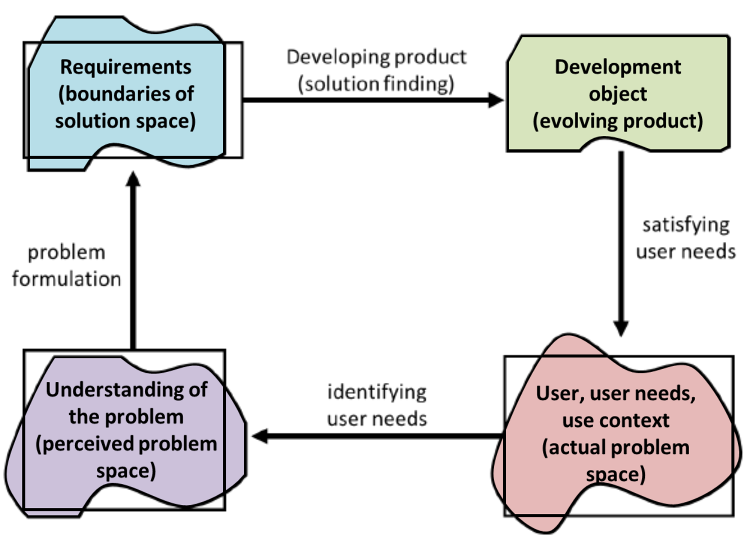
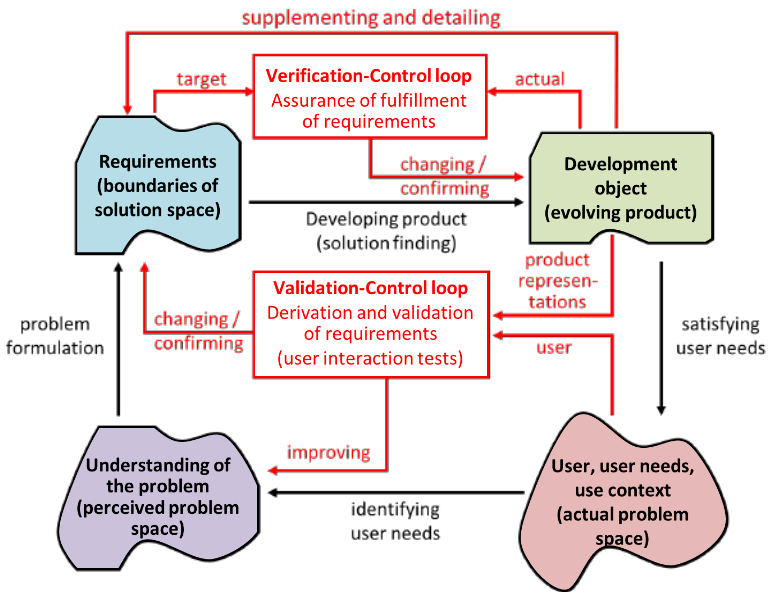


Figure 3 Validation and verification control loop of an agile product development process (see online version for colours)



Source: cf. Koppenhagen et al. (2021b)

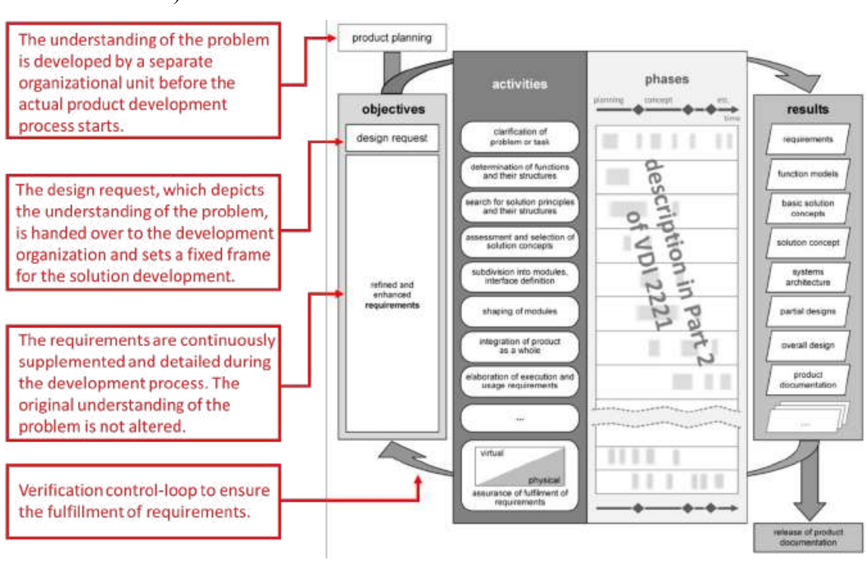
The validation control loop is established between the development object and the actual user need. Right from the start, user interaction tests are carried out, in which prototypes of different scope and resolution represent different aspects of the emerging product. With these tests an understanding of user needs develops, and requirements can be derived and validated (Koppenhagen et al., 2021a, 2021b). The validation control loop determines the user-centered evolution of problem understanding and thus defines the basic direction of the development efforts. It prevents a mismatch between the development result and the user needs. The verification control loop, on the other hand, is established between the requirements and the development object and assures the fulfilment of the defined requirements (Koppenhagen et al., 2021a, 2021b). For this purpose, physical tests as well as numerical simulations are utilised. The verification loop controls the technical realisation, i.e., the concrete solution development in the solution space.

In principle, both the underlying understanding of the problem and the boundaries of the solution space remain volatile during an agile development process. However, in practical application, the validation control loop usually dominates the early phase of the development process. After the validation control loop has settled, the focus shifts to the verification control loop. The verification control loop ultimately brings about a convergence of the solution development in the solution space. This is done through the continuous addition and detailing of the requirements.

The dual control loop makes the agile development process inherently more adaptive than a conventional systems engineering development process with only a single control loop as described below and fosters the user-centered generation of radical innovations. The practical implementation of agile development includes implications for the design of the corporate organisation though. The co-evolution of problem and solution space requires responsibility for the development of problem understanding and solution to be unified both organisationally and personally. In other words, the people responsible for understanding the needs and use contexts of potential users must also be responsible for developing the solution (Koppenhagen and Wecht, 2023). Separating the responsibility for understanding the problem and developing the solution, as is common in the operational practice of most manufacturing companies today, prevents the establishment of the validation control loop and thus also the implementation of the agile development process.

In contrast to agile development, conventional systems engineering is characterised by a clear separation between the problem understanding and the solution development phases, which are run through sequentially. This is reflected in both the VDI 2221 (2019) and the systematic development approach of Pahl et al. (2007), two established process models from systems engineering. While the development of the underlying problem understanding is still part of the original engineering design methodology of Pahl et al. (2007), it is upstream of the actual development process according to VDI 2221 (2019). In the VDI 2221 (2019), the underlying understanding of the problem is developed in a pre-project phase, the so called product planning phase, the result of which is finally handed over to the development organisation as a design request (Figure 4). In industrial practice, the product planning phase is usually the responsibility of an organisational unit with a direct market or customer interface (e.g., product management, sales, etc.), which is normally not involved in subsequent solution development. The design request comprises socio-demographic information on the target customer segment and formulates the product's performance profile necessary to achieve the intended strategic positioning relative to existing competitive products. It describes, at a high level of aggregation, requirements for the product to be developed and already implies certain solution principles and concepts for their fulfilment, thus creating a fixed framework for subsequent solution development.

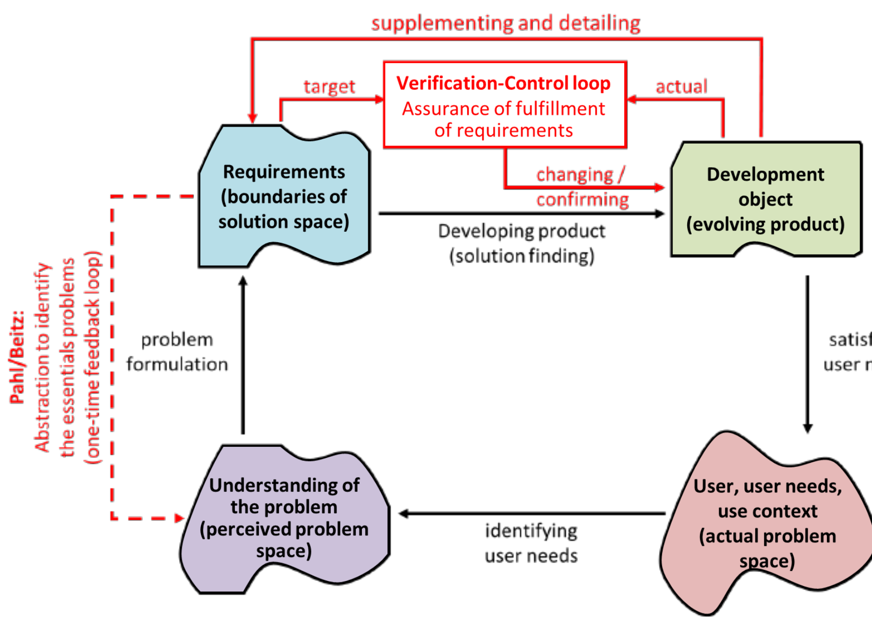
Figure 4 The general model of product design according to VDI 2221 (2019) (see online version for colours)



Authors' additions in red.

In principle, changing the underlying problem understanding during solution development is not intended in the VDI 2221 (2019). In the first step of this process model, the problem understanding delineated in the design request is operationalised for further solution development. This happens by clarifying specifications and detailing requirements. During actual solution development, only a verification control loop is established between the requirements and the development object to assure the fulfilment of requirements (Figure 5). The process model does not include a control loop for validating the requirements, i.e., for checking whether these requirements reflect the actual user needs.

Figure 5 Verification control loop of conventional systems engineering and abstraction to identify the essentials problems according to Pahl et al. (2007) (see online version for colours)



In the design methodology of Pahl et al. (2007), the problem formulation depicted by the requirements is critically reviewed at least once before the start of solution development. This is done through a systematic abstraction and broadening of the problem formulation to identify the essential problems at the start of the conceptual design phase. It should be checked, „*if an extension of, or even a change in, the original task might lead to promising solutions.*" (Pahl et al., 2007). The abstraction of the problem formulation should help designers to abandon cognitive solution fixations and conventions that might have been included in the requirements while also preventing the solution space from being prematurely narrowed down by an inadequate problem formulation.

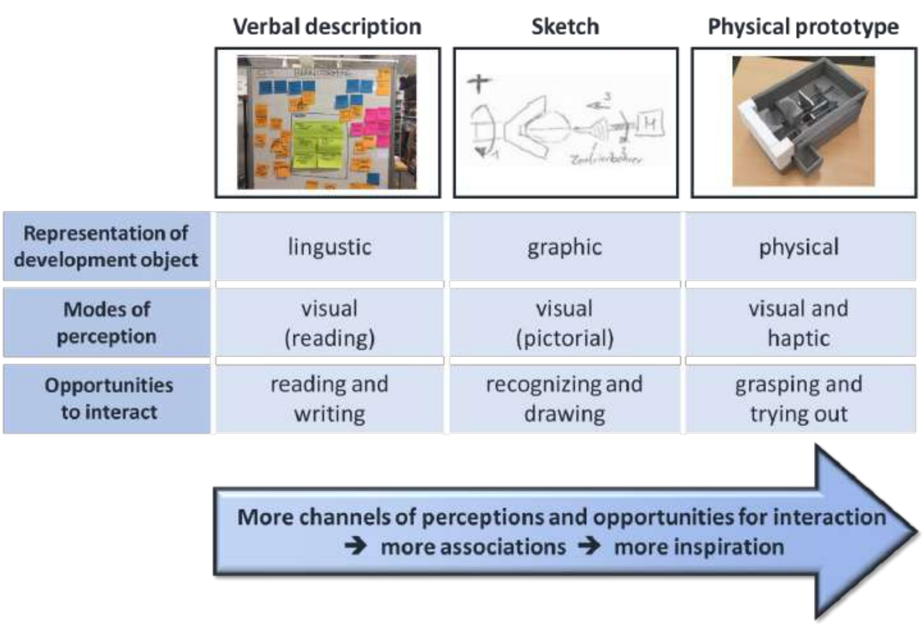
Product development process models separating problem understanding and solution development are '*specification-driven*' (Reinertsen, 1997) and strongly focus on convergence. The strength of these approaches lies in the systematic solution development for well-defined problems. They face limitations, however, when it comes to radical innovations, the development of which normally requires a new understanding of the underlying problem and the user needs associated with it. Since problem space and solution space are not separate spheres but two sides of the same coin, a comprehensive and deep understanding of the problem often only emerges during solution development. Radical innovations therefore usually result from a user-centered iterative co-evolution of problem and solution space, as is characteristic of agile product development approaches.

2.2 Implications of the agile development of physical products: the need for a hybrid process model

Carrying out user interaction tests within the validation control loop described in Section 2.1 requires respective product representations. In the development of physical products, these product representations consist of tangible prototypes. Agile product development processes are therefore always prototype-based development approaches.

The use of prototypes brings numerous advantages for the process of creative solution development. A physical prototype enables three-dimensional visual and haptic perception and provides designers or potential users with the opportunity for tangible interaction. Opposed to a graphic or verbal representation of ideas, a prototype stimulates significantly more perceptual channels (McKim, 1980; Edelman et al., 2009; Leifer, 2012). This way, the prototype can evoke significantly more associative links among development team members than virtual product representations, which in turn can lead to more and better ideas (Figure 6). In particular, the unfinished character of low resolution prototypes, leaving room for ambiguous interpretation, can stimulate generative discussions often leading to a fundamental change rather than just an incremental improvement of the embodied ideas (Brereton and McGarry, 2000; Edelman et al., 2009).

Figure 6 Connection between development object representations and inspiration of ideas (see online version for colours)



Prototypes also reveal the real-world properties and limitations of physical solutions (Brereton and McGarry, 2000). They are able to correct flaws in the designers' mental models and contribute to a better understanding of the components and the existing boundary conditions and restrictions (Viswanathan and Linsey, 2012; Böhmer et al., 2017). As a result, prototypes lead to more technically feasible ideas in creative development processes (Viswanathan and Linsey, 2012).

In contrast to technical drawings, 3D CAD models or numerical simulations, physical prototypes are also able to represent qualitative elements that make a product meaningful to a user (Edelman et al., 2009). In direct user interaction, they often lead to unexpected discoveries from which requirements can be derived, some of which only become apparent in practical testing (McKim, 1980).

Prototypes also improve both internal and external team communication. They embody ideas concretely and make them easier to understand for team members, potential users, and industry partners, thus preventing unnecessary misunderstandings (Berglund and Leifer, 2013; Böhmer et al., 2017). They continuously represent the maturity level of an emerging product with respect to different disciplines providing all members of a multidisciplinary team with clarity about the project status (Böhmer et al., 2017). Furthermore, prototypes convey a vision of the later product, which supports the development of a shared project vision in the development team (Böhmer et al., 2017).

With respect to solution space exploration, a prototype-based development approach is a double-edged sword. In contrast to software engineering, with physical products, development and manufacturing are fundamentally separate processes. Building physical prototypes involves significantly more effort than pure software development, especially for complex products. In addition, for physical products, interactions between product components and within the human-product system are not limited to the signal flow, i.e., to logical relationships, but include all interactions that can result from a flow of forces, energy and materials. This means that testing and evaluating physical prototypes also involves more effort than testing software versions.

Since building and testing physical prototypes is associated with considerable effort and time, development on the overall system level can rarely be executed in a completely agile manner, especially for complex products. Process models that are to be used in the context of development projects with limited time and resources must therefore guide the convergence of solution development. For this, two essential milestones must be defined with respect to (1) the development of problem understanding and (2) solution development:

1. a milestone, from which the underlying understanding of the problem is no longer changed but is set as a fixed framework for further solution development
2. a milestone for concept selection, i.e., a principle solution on the overall system level before complex system prototypes are built.

It should be noted that these milestones relate to problem understanding and solution development at the overall system level. For certain subsystems, e.g., specific physical subsystems or software scopes, development can still be carried out in an agile manner after these milestones.

The two essential milestones separate the development process at the overall system level into an agile and a convergent development phase. Such a separation is also the prerequisite for a stable process-related and contractual integration of suppliers into the development process. Process models that provide this separation between an agile and a convergent development phase are referred to as hybrid process models in the context of this paper. In hybrid product development process models, there is thus an inherent tension field between agility and convergence in solution development.

The greater effort associated with prototype-based solution development also aggravates a tension field that generally exists in NPD: the polarity between parallel and iterative solution development, i.e., between the breadth and depth of the solution space exploration. Building and testing a physical prototype does generate greater gains in knowledge with regard to the specific embodied solution, thus enabling deeper solution space exploration than a purely virtual development. In the case of complex products, however, parallel construction and evaluation of competing solutions quickly leads to prohibitive effort, particularly at the overall system level. Although Dow et al. (2010) have shown that parallel prototyping leads to better development results, agile development approaches therefore tend to be characterised by an iterative '*point-based*' (Sobek et al., 1999) solution development. This selectively deep but less broad exploration of the solution space increases the risk of finding only a local optimum.

In contrast to the prototype-based solution development described in the previous paragraph, the early phase of solution development in systems engineering is usually characterised by greater breadth but less depth. In systems engineering, solution development is primarily virtual in the early phases, i.e., absent of physical prototype building. At least until the concept is selected, this allows for parallel development, enabling the elaboration and evaluation of competing solutions both on the subsystem and the overall system level. Moreover, the decomposition of the overall system being characteristic for systems engineering allows for a systematic variation and combination of solution components, facilitating the theoretical exploration of a large solution space. Only from the moment of concept selection onward does further development of the solution follow a point-based approach. However, the largely virtual development leads to less knowledge gain when generating and evaluating solutions compared to prototype-based approaches, which in turn increases the uncertainty in design decisions.

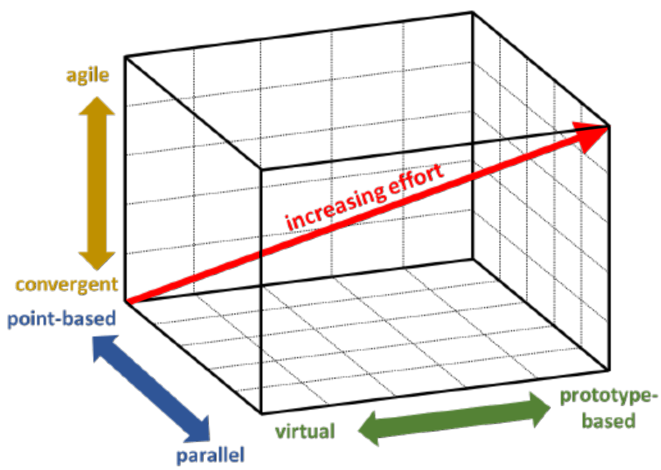
The adequate positioning in the three-dimensional tension field spanned by the polarities of

1 agile and convergent development,
2 parallel and point-based development and
3 prototype-based and virtual development, is the central challenge in designing a hybrid product development process model and defines its practical suitability (Figure 7).

2.3 Semantic model of design thinking

Although several attempts have been made to develop a consistent theoretical frame of reference for design thinking (cf. e.g., Carlgren et al., 2016; Hassi and Laakso, 2011), the understanding of design thinking still varies greatly depending on the academic perspective and practical application context. Design thinking is often understood as a generic reference for an innovation process that is particularly user-centered and in which the participants consider certain ways of thinking, attitudes, and working principles as action-guiding paradigms, which, at the same time, also express a certain philosophy and culture (Gruber et al., 2015). In some cases, reference is made to individual techniques and methods that are used in the context of design thinking (cf. e.g., Gerstbach, 2017), but which do not originate from design thinking and are also frequently used in other contexts. Thus, design thinking represents a rather loose umbrella term that can have various meanings (Johansson-Sköldberg et al., 2013; Kimbell, 2011). The articulated concepts, however, usually remain vague. Even people claiming to use design thinking regularly in business practice often have difficulties explaining their understanding of it (Carlgren et al., 2014). In the academic discourse on design thinking, the very question of the need for a consistent definition is controversial: While some scholars, such as Badke-Schaub et al. (2010), consider a consistent definition of design thinking a prerequisite for systematic research on the concept, others, such as for example Johansson-Sköldberg et al. (2013), regard the attempt to establish a "*unique meaning of design thinking*" (Johansson-Sköldberg et al., 2013) as an improper simplification and narrowing of design thinking's multi-perspective and multi-facetted approach. Even the term Design Thinking itself is partly controversial among proponents and users of the concept (Liedtka, 2015).

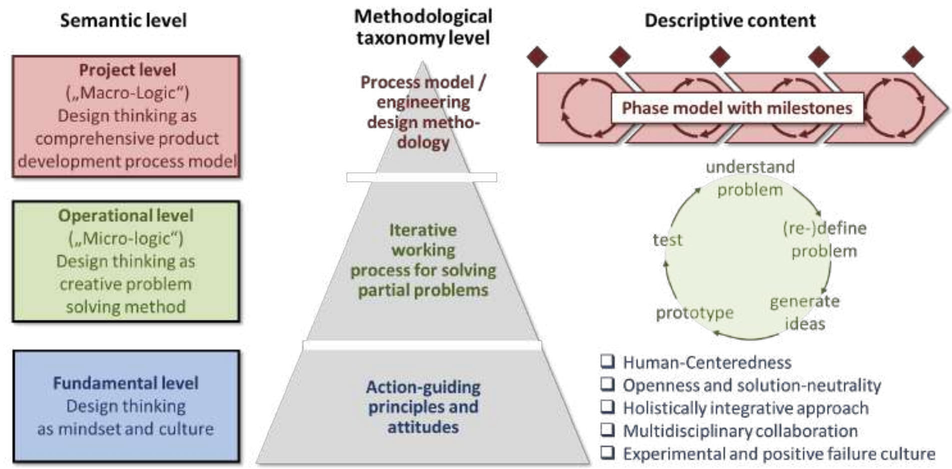
Figure 7 Three-dimensional tension field of hybrid product development (see online version for colours)



The lack of a consistent definition of design thinking makes it difficult to classify the concept with regard to engineering design methodology. This is one reason why design thinking has thus far received little attention in the academic discourse of engineering design methodology. The limited number of studies analysing design thinking in terms of engineering design methodology, such as in Gericke et al. (2010) and Schüttoff et al. (2019), mostly compare different reference levels: on the systems engineering side, a design methodology and process model for structuring an entire product development process, with; on the design thinking side, an iterative working process for solving partial design problems. This is also due to the fact that most design thinking process models do not encompass an entire product development process.

This paper's understanding of design thinking entails a model comprising three semantic levels that are hierarchically tiered. These semantic levels also represent different methodological taxonomy levels within the context of product development (Figure 8).

Figure 8 Semantic levels and methodological taxonomy levels of design thinking (see online version for colours)



The fundamental level describes the action-guiding principles that are constitutive for the mindset and culture of the design thinking approach (cf. e.g., Carlgren et al., 2016). The operational level illustrates the core cycle of design thinking, an iterative working process representing the operational implementation of the fundamental level's action-guiding principles. From a design methodology perspective, the aforementioned core cycle is a process designated for solving partial problems. This level is also referred to as 'micro-logic' by Haberfellner et al. (2019). For the Design Thinking´s core cycle, different representations and descriptions can be found in the literature, in which both the number and the naming of the individual process steps vary. Nevertheless, said representations refer to an almost identical process at their core (Schüttoff et al., 2019).

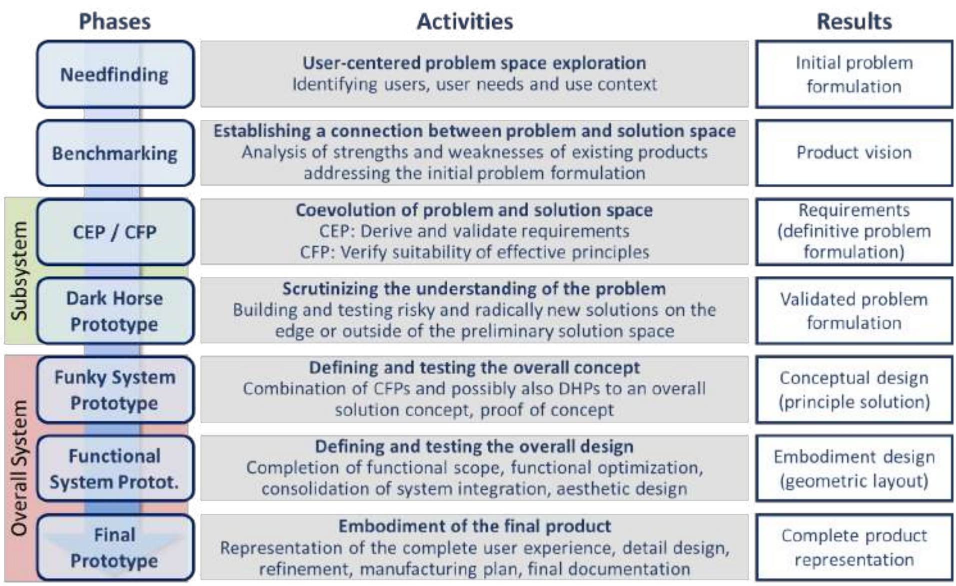
The process models or, respectively, the phase models, both of which structure a complete product development process are located on the third and final level, the project level, also referred to as 'macro-logic' by Haberfellner et al. (2019). In contrast to the operative core cycle, the phase models are only carried out once throughout the course of an entire project. Design thinking process models must, on the one hand, transfer the fundamental principles of design thinking to the higher project level, structuring the overall process and integrating the iterative core cycle for solving partial problems within respective phases. On the other hand, these process models must also guide convergence during solution development on the overall system level with suitable milestones. Said milestones must, in turn, synchronise participants' cooperation in the product development process, providing them orientation on the project stage that enables them to derive not only the tasks to be performed but also the degrees of freedom remaining within their area of responsibility. An extensive literature review indicates that the only process model meeting these criteria is the ME310 process model, developed at the Center for Design Research at Stanford University, which is described in the next section.

2.4 Stanford´s ME310 process model

ME310 is a project-based graduate course in which a Stanford University student team collaborates with a foreign partner university's team to develop innovative products (ME310 refers to the course's catalogue number; for an overview of the roots and history of the ME310 course, see Carleton and Leifer (2009) and Carleton (2019)). The project prompts comprise real-world design challenges from cooperating industrial companies. Taking place over three quarters, the course's duration translates to a total of thirty weeks. Both teams are supervised by professors, lecturers and course assistants.

ME310 is also the name of the process model according to which students work on their development projects. The following description of the process model's individual phases along with their assigned activities and intended results, is based on the ME310 ABC Course Reader (Kenyon et al., n.d.) (Figure 9).

Figure 9 Phases, activities and results of the ME310 process model (see online version for colours)



Source: Koppenhagen et al. (2021b)

2.4.1 Needfinding (NF)

In the Needfinding phase (NF), starting from the project prompt, different ethnographic methods are used to identify the users, their needs and problems, and the context of use. This includes in particular user interviews and observations, but also putting oneself in the position of the user. The information obtained is illustrated and summarised by sketching an archetypal user according to the persona concept. Based on the model of the value proposition canvas (Osterwalder et al., 2014), the most important 'customer jobs', existing 'pains' and possible 'gains' are formulated. The result of this phase is an initial problem formulation, which represents the starting point for the subsequent development process.

2.4.2 Benchmarking (BM)

The Benchmarking phase (BM), in which the strengths and weaknesses of existing products in the defined problem space are analysed, already establishes a connection between problem and solution space: on the one hand, this analysis improves the understanding of the problem, as it becomes evident which problems have not yet been satisfactorily solved. On the other hand, it also provides initial insights into the solution space, since the structure of the products, the technical function carriers and their effective principles to fulfil the functions become apparent. The Benchmarking thus shows opportunities for innovative differentiation from existing products and, at the same time, provides initial inspiration and starting points for subsequent solution finding. By combining the results of the Needfinding and Benchmarking phases, an initial product vision develops, which includes a strategic positioning of the product to be developed.

2.4.3 Critical experience prototype (CEP) und critical function prototype (CFP)

In the ME310 process model, actual solution development begins in the CEP/CFP phase at the subsystem level. Critical experience prototypes (CEP) are built to make critical core elements of the user experience from the product vision tangible. CEPs facilitate an understanding of the problem space, which helps to derive and validate user requirements. This is often done by using "Wizard of Oz"-prototypes, where the user experience of a function is simulated without having already developed a technical function carrier. Critical function prototypes (CFP), on the contrary, help to evaluate the suitability of effective principles for selected function carriers critical to the overall concept's technical solution. Several CEPs and CFPs are built and further developed iteratively within this phase. The CEP/CFP phase concludes with the formulation of functional and physical requirements, which encompass both a definitive problem formulation and a "coherent vision" (Domingo et al., 2020) of the product to be developed, thus combining desirability with technical feasibility.

2.4.4 Dark horse prototype (DHP)

The dark horse prototype (DHP) phase is intended

1. to validate the acquired problem formulation, which, in other words, depicts the understanding of the problem
2. to prevent the solution space from being prematurely narrowed down.

For this purpose, prototypes are built that involve a particularly risky, radical or unconventional solution principle, perhaps initially regarded as infeasible within the CEP/CFP phase (Bushnell et al., 2013). The creation of DHPs forces the development team to abandon an underlying cognitive solution fixation (Domingo et al., 2020) and scrutinise previous understandings of the problem. Dark horse prototypes can lead to both a reorientation in the solution space as well as to a change in the underlying problem formulation. This phase should result in a validated problem formulation, generating a firm and reliable framework for subsequent solution development on the overall system level.

2.4.5 Funky system prototype (FKP)

The Funky system prototype (FKP) is the first system-level prototype to define the overall concept and ensure its suitability. For this purpose, the most promising function carriers from the CFP phase (and, if applicable, the Dark Horse phase) are to be combined to form an overall solution. The FKP rarely represents the complete functional scope, rather concentrating on the solution-determining main functions; its only purpose is to technically verify the effective structure of the overall solution. Formal aesthetic design features do not yet play a role in the FKP.

2.4.6 Functional system prototype (FCP)

In the functional system prototype (FCP) phase, the concept of the FKP is detailed and optimised. The FCP is intended to represent the complete functional scope and serves to consolidate system integration as well as optimisation on an overall and subfunction level. It defines the embodiment design and should already have a value proposition comparable to the final prototype.

2.4.7 Final prototype (FP)

The final prototype, marking the completion of development, should represent the complete user experience of a product to be industrially realised. The main focus in this project phase is on detail design and refinement to ensure a high-quality development result. In addition, the manufacturing plan and the final product documentation are drawn up in this phase.

The additional Part-X-is-finished prototype listed in some publications (e.g., Domingo et al., 2020) refers to the completion of a student's component of choice in the Final Prototype phase. This physically realised design freeze, which only refers to one specific component, is intended to break the cycle of mutual geometric structural dependencies existing in a product architecture and marks the crystallisation of the Final Prototype. One can therefore regard the Part-X-is-finished prototype as an intermediate milestone within the Final Prototype phase.

3 Investigation of the ME310 process model

3.1 Research methodology

In this section we assess the ME310 process model's usefulness with respect to the development of physical products. The fundamental nature of engineering design complicates the assessment of design methodologies considerably. Since Engineering Design deals with problems that have an open problem and solution space, there is no single right solution to an engineering design process; instead, several solutions are conceivable. The evaluation of these solutions not only involves objective, but also subjective criteria related to the different perspectives and needs of the various stakeholders. Thus, it is sometimes even difficult to agree on common criteria for assessing the development result. However, if a purely objective evaluation of a design solution is not possible, it is also not possible to assess the design methodology applied to develop this solution through a formal, rigorous and quantifiable scheme (cf. Pedersen et al., 2000). We have therefore chosen a qualitative approach that combines a theoretical and an empirical analysis.

In the theoretical analysis in Section 3.2, we first investigate the '*structural soundness*' of the process model, i.e., it´s '*internal consistency*' according to Pedersen et al. (2000). For this purpose, we analyse the individual process steps and their connections and compare them with the established process models of systems engineering, i.e., the VDI 2221 (2019) and the development methodology according to Pahl et al. (2007). In particular, we examine whether "*for each step* […] *there is adequate input available, that the anticipated output from the step* […] *is likely to occur based on the input, and that the anticipated output is an adequate input to another step*" (Pedersen et al., 2000).

In the subsequent empirical analysis in Section 3.3, we investigate the practical suitability of the ME310 process model for developing physical products. Since it aims to combine agility and convergence in the realm of NPD, we wanted to determine whether, in practice, convergence during solution development corresponds to the ME310 process model's theoretical specifications. Therefore, using a qualitative case study approach, we analysed 10 of Stanford's ME310 students' product development projects. In these projects, students developed physical products in response to real-world design challenges coming from industry sponsors. Within the scope of our empirical analysis, we examined the technical concepts of 177 prototypes and their underlying problem formulations in detail. Thus, we were able to reconstruct the actual coevolution of problem and solution space and compare it with the theoretical specifications of the ME310 process model. From this comparison, conclusions about the process model's practical suitability for the development of physical products can be drawn. If the development processes observed in practice deviate significantly from the process model's theoretical specifications, this can be seen as an indication of its limited practical suitability for the development of physical products. For then, it was apparently either not possible or not beneficial to achieve the development result in the way specified by the process model, because certain process steps do not have adequate input or are not suitable for providing the desired output for subsequent process steps on the basis of their existing input.

By comparing and relating the results of the theoretical and empirical analysis, we can finally derive specifications for the development of a new hybrid process model with high practical suitability for the development of physical products.

3.2 Theoretical analyses of the ME310 process model

Our analysis of the ME310 process model and its comparison with the design methodologies of Pahl et al. (2007) and the VDI 2221 (2019) is structured in such a way that we examine the fundamental principles of the respective design methodologies along the following five polarities:

i Agility vs. convergence in solution development

ii Parallel vs. iterative solution space exploration

iii Prototype-based vs. virtual development

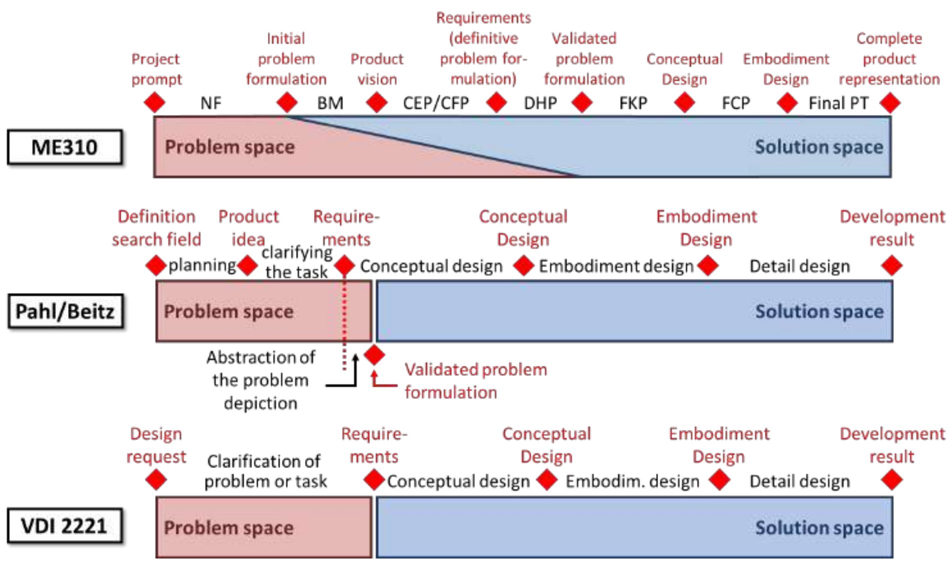
iv Overall system level vs. subsystem level in solution development

v Internal functional relationships vs. human-product-interaction

i Agility vs. convergence in solution development

While the systems engineering approach usually clearly separates problem analysis and solution development (see also Section 2.1), the ME310 process model is characterised by a concomitant development of these two spheres in the early prototype stage (Figure 10). The two milestones being constituent for hybrid development process models, i.e., the final definition of (1) the understanding of the problem and (2) the solution concept on the overall system level, are marked in the ME310 process model by completion of the phases (1) Dark Horse Prototype and (2) Funky System Prototype, respectively.

Figure 10 Development of problem and solution space in the ME310 process model, the systematic approach of Pahl et al. (2007) and the VDI 2221 (2019) (see online version for colours)



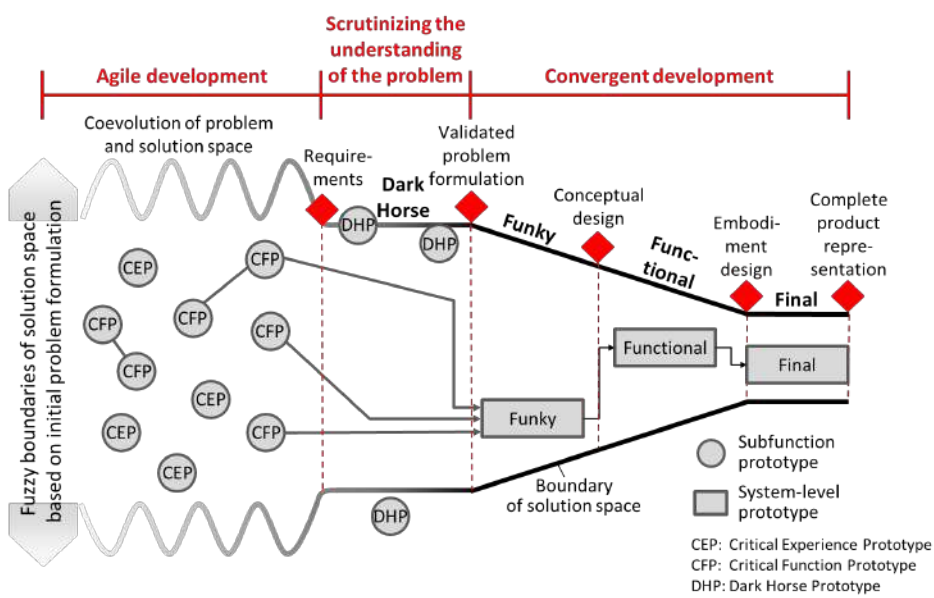
In the ME310 process model, development is only completely agile during the subfunction prototype phases (CEP/CFP and DHP), in which the understanding of both the problem and, consequently, the boundaries of the solution space remain volatile (Figure 11). The validation and verification control loop, which in their interaction control the agile product development process (see Section 2.1), are hereby established by the CEP and the CFP, respectively:

The user interaction tests based on the CEPs deepen the understanding of the problem and enable the derivation and validation of requirements. The CEPs thus establish a validation control loop that prevents a 'mismatch' between product features and user needs. At the same time, the CFPs ensure the suitability of innovative effective principles and the fulfilment of requirements for the solution-determining main functions; they thus establish a verification control loop between the requirements and the development object. Through an iterative development process guided by a stringent user-centered approach, the CEPs' and CFPs' interaction implements the principle of the problem and solution space's coevolution. Thus, the CEP/CFP phase of the process model shows a methodological self-similarity to the design thinking's iterative core cycle.

In contrast to systems engineering, the problem formulation in the ME310 process model, meaning the formulation of requirements depicting and operationalising the understanding of the problem, does not occur before the start of solution development but rather after solution development completion at the subsystem level. Similar to the original design methodology of Pahl et al. (2007), this understanding of the problem is first critically reviewed before it is set as a fixed framework for further solution development. In contrast to Pahl et al. (2007), this is not accomplished through a theoretical abstraction of the problem formulation, but by building and evaluating concrete solutions on the edge or outside of the preliminary boundaries of the solution space using DHPs. Both approaches nonetheless pursue the same goal: avoiding a premature and overly narrow limitation of the solution space through an inadequate problem formulation.

After completion of the DHP Phase, a solution concept is developed in the FKP Phase based on the validated problem formulation and finally selected upon completion of this phase. The solution concept is then further developed into a complete overall design in the subsequent phase of the FCP and finally detailed in the final prototype phase. In the system prototype phases, thus, an increasing convergence and consolidation of the development result is sought, which, as in systems engineering, is controlled by supplementing and detailing requirements, leading to an increasing solution space limitation. The deductive solution development on the overall system level in the ME310 process model strongly resembles the development methodology according to Pahl et al. (2007) and VDI 2221 (2019). In all three of these design methodologies, a principle solution is first developed and then elaborated on, resulting in an overall design defining the geometric layout of the product.

Figure 11 Theoretical convergence path of the ME310 process model (see online version for colours)



ii Parallel vs. iterative solution space exploration

In contrast to systems engineering that normally involves parallel solution development both at the overall and at the subsystem level until the concept is selected, parallel development in the ME310 process model is only intended for the level of subfunction prototypes. On this level, several CFPs are built in parallel to evaluate different effective principles for fulfilling the solution determining main functions of a possible concept, with the different CFPs in turn being further developed iteratively. Parallel construction and evaluation of competing concepts at the overall system level, i.e., parallel prototyping in the FKP phase, is not intended. Although the FKP only represents a principle solution and still has a comparatively low resolution, the effort involved in building different concepts on the overall system level is still prohibitive, especially in the case of complex products. In this respect, the approach of the ME310 design methodology is, like all prototype-based approaches, even more point-based than the traditional systems engineering design methodology (cf. Section 2.2). The solution space explored at the time of concept decision is smaller overall, increasing uncertainty in concept selection.

iii Prototype-based vs. virtual development

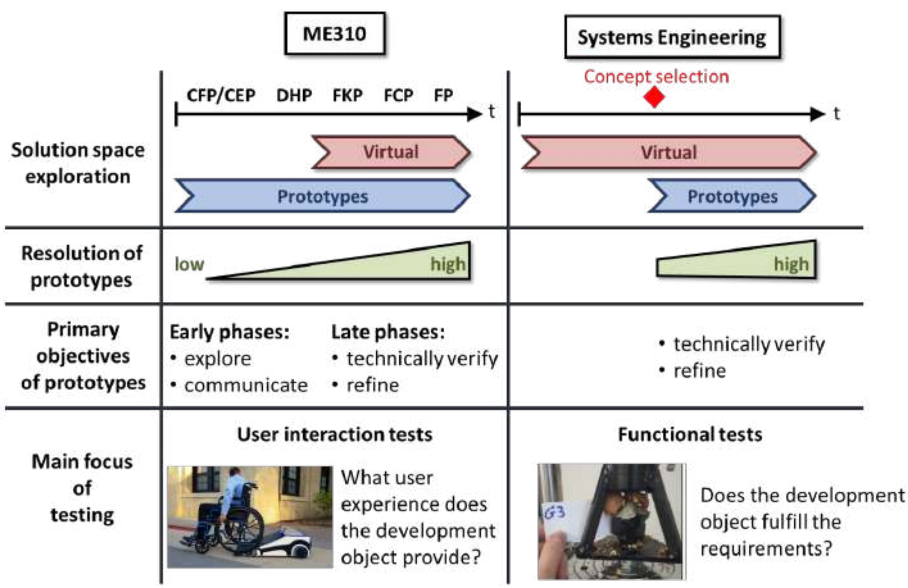
In this section, the use of prototypes and the role of virtual development in the ME310 process model will be examined and compared with systems engineering. This paper's understanding of virtual development includes all non-physical product representations. Said representations include not only computer-aided geometry, calculation, and simulation models, but also manually created drawing and sketches, for example. Prototypes on the contrary are understood as physical objects that represent certain functions or properties of an evolving product (cf. Lauff et al., 2017; Otto and Wood, 2001). Prototypes are used to increase knowledge about the development object, thus reducing uncertainties in design decisions. According to Grauvogl (2018), the following objectives of prototypes can be distinguished (cf. also Camburn et al., 2017; Hallgrimsson, 2012):

- **Explore:** The prototype supports problem space exploration, i.e., it helps to better understand the problems and needs of potential users, and to guide the basic direction of development. Regardless of the specific technical implementation, it should answer the question of *what* should be developed. This is done through user interaction tests, where the prototype enables requirements to be derived and validated.
- **Communicate:** The prototype helps to communicate and explain ideas and their degree of maturity to potential users, company partners or members of the development team.
- **Technically verify:** The prototype is used to verify the fulfilment of requirements by the function carriers. It helps to evaluate concrete technical solutions, e.g., to assess the suitability of effective principles. This is done through functional tests, i.e., physical quality assurance. The focus is on answering the question of *how* something is to be realised.

- **Refine:** The prototype aims to refine an idea that has already been implemented, usually with the aim of incrementally optimising certain often qualitative product properties. In a refinement, the resolution of the prototype, i.e., its level of detail and degree of maturity, is usually increased.

In systems engineering, the use of physical prototypes is not a constituent part of the design methodology, but only a means of property assurance during the development process. In order to save costs and time, development remains virtual for as long as possible, especially in the case of high product complexity. This includes not only the geometric description of the product, but also the simulation of almost all product functions and properties as part of virtual quality assurance. Even though certain design decisions in early development stages, such as, e.g., determining the formal appearance of a product, are partly supported by physical product representations, prototypes are usually only used in later development stages, when digital development is almost complete, i.e., when most design decisions have already been made. These prototypes generally already have a high resolution and are used to validate the results of virtual quality assurance or to ensure the fulfilment of product properties that cannot yet be assured virtually (Figure 12).

Figure 12 Prototype based vs. Virtual development in the ME310 process model and systems engineering (see online version for colours)



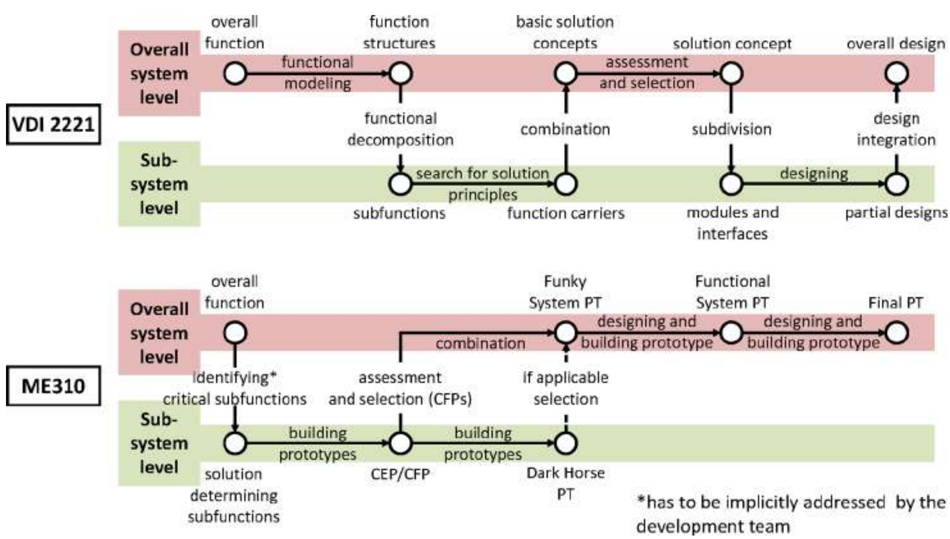
In contrast to systems engineering, the building of prototypes is integral to the ME310 design methodology; they are the main drivers of the development process. Within all phases, prototypes are used not only for verification and refinement but also for exploring and communicating solutions. The different kinds of prototypes define the process model's milestones, which structure the overall development process and guide convergence of solution development.

Early subfunction prototypes (CEPs, CFPs and DHPs) are usually built within the ME310 process without prior virtual development. They are often created through practical improvisation. Intensive virtual development usually starts before system prototypes (FKP, FCP and FP) are built, as their increased complexity necessitates virtual development prior to their construction.

iv Overall system level vs. subsystem level in solution development

The entire solution development process according to VDI 2221 (2019) is characterised by a permanent oscillation between the overall and the subsystem level, i.e., a continuous alternation between a top-down and a bottom-up approach (Figure 13). At the beginning, a functional modelling of the overall system takes place through establishing function structures, which mark the starting point for explicit functional decomposition. Actual solution finding then starts at the subsystem level, at which effective principles and function carriers for the fulfilment of the identified subfunctions are sought. This discursive process is usually supported by the use of classification schemes (e.g., morphological chart). By systematically varying and combining different effective principles, several solution concepts are then generated at the overall system level, from which one solution concept is finally selected by means of a technical and economic evaluation. The selected concept is subdivided into modules which, after appropriate definition of the interfaces, represent the starting point for developing the embodiment design. The partial designs for the individual modules are finally integrated into an overall design.

Figure 13 Solution development between overall and subsystem system level in VDI 2221 (2019) and the ME310 process model (see online version for colours)



In the practical implementation of the ME310 process model, ideas for possible overall solutions are generated at the beginning of the CEP/CFP phase. Usually, the team of developers use brainstorming for this purpose. In principle, however, other intuitive or conventional solution methods can also be used. The generated ideas are often characterised by a particularly innovative effective principle for one or more solution-determining main functions. The identification of these solution-determining main functions takes place in free discussion without methodological support by the process model.

In addition, the ME310 process model does not provide any methodological support for the design of the overall system. What, in particular, does not occur at all is the functional modelling of the overall system, i.e., the establishing of function structures; and this has two major implications. First, the basis for an explicit functional decomposition is missing. Second, it complicates the analysis of functional relationships, respectively technical interactions between function carriers in the overall system. Such an analysis is, however, crucial to the development of a technical concept. Therefore, no methodological support exists for:

1 the identification of the product vision's solution-determining elements that should be embodied during the CEP/CFP phase

2 the identification of compatible combinations of different CFPs to form the first overall solution, the FKP. Both have to be addressed implicitly by the development team, a substantial challenge, especially for complex products.

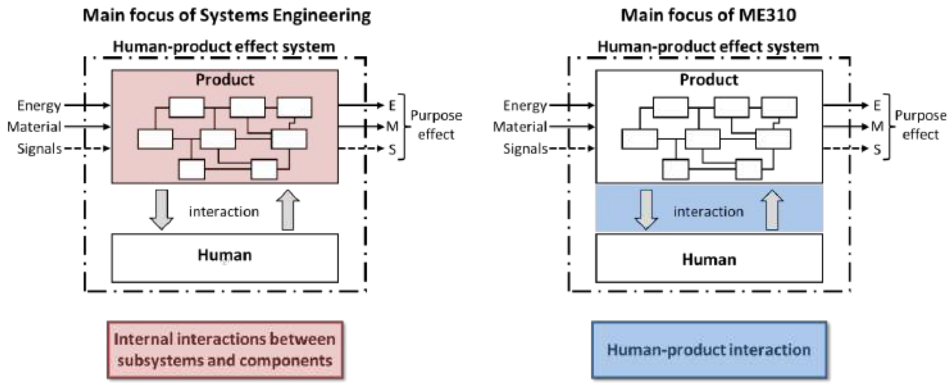
Indeed, the modelling of an overall system and its decomposition into subsystems is crucial for handling complexity because it is an indispensable prerequisite for breaking down the development task into manageable subtasks (Kersten and Koppenhagen, 2002; Koppenhagen, 2011). The lack of such an explicit approach to complexity reduction limits the application possibilities of the ME310 development methodology.

A development of different solution concepts and their systematic evaluation finally leading to a concept selection is not part of the ME310 process model. Instead, after creation of the FKP, the intended goal of the process model is only a continuous extension of the functional scope and an increase in resolution leading to the Functional System. Since the FKP is the first system prototype that can be used to test and analyse the interaction of the different function carriers, this approach seems overly optimistic, especially for complex products.

v Human-product interactions vs. system design

Even though the development methodologies according to Pahl et al. (2007) and the VDI 2221 (2019) both start by describing the overall function of a product to be developed, which also includes the human-product interactions, the focus of development is clearly on the system design, i.e., on the internal interactions of the various subsystems and components. This applies to all stages of the development process. Both in the creation of function structures and effective structures, the analysis and design of the functional interactions between different components is the focus of the development efforts. This continues in later development stages with the design of the components and the modules with their respective interfaces. Both the design methodologies according to Pahl et al. (2007) and the VDI 2221 (2019) are thus characterised by a pronounced introspection (Figure 14). There are no milestones that explicitly refer to the human-product interactions. Optimising usability and improving the user experience is neither a main focus of the design methodology according to Pahl et al. (2007) nor the VDI 2221 (2019).

Figure 14 Human-product interaction vs. system design in systems engineering and the ME310 process model (see online version for colours)



In this respect, the ME310 design methodology clearly stands in contrast to the established design methodologies from systems engineering. From the outset, the design of the human-product effect system is the central guiding paradigm of its development efforts. The focus in all prototype phases is on testing user interaction to evaluate and improve usability and the user experience of the evolving product. The key elements of the user experience are investigated with CEPs before technical solutions are even developed to fulfil the respective functions. The ME310 design methodology is thus characterised by a user-centred approach generally characteristic of design thinking. The system design with the analysis and design of the interactions between subsystems and components is done almost incidentally and is not explicitly described methodologically. The conclusions that design thinking is not suitable for the development of '*very technical issues*' (Schüttoff et al., 2019) or of products that "*have no user interaction at all*" (Gericke et al., 2010) therefore seem plausible.

3.3 Empirical analyses of the ME310 process model

3.3.1 Research questions for the empirical analyses

The objective of the ME310 process model's empirical analysis explained in Section 3.1 is operationalised by the following research questions:

1. How many of the different prototypes were built on average in the development projects?
2. Which objectives were associated with the different prototypes?
3. What percentage of the different prototypes were tested with external users?
4. What percentage of the different prototypes were built in parallel?
5. Did the development of the problem formulation follow the theoretical specifications of the process model?
   a. How often and in which project phases did changes in the underlying understanding of the problem occur?

b In which project phase was the Final Prototype's underlying problem formulation defined?

c Were the changes in the underlying understanding of the problem triggered internally by members of the development team or externally by the teaching team or representatives of the cooperating industrial company?

6 Did the concept development follow the theoretical specifications of the process model?

a How often and in which project phases did concept changes occur?

b In which project phase was the Final Prototype's concept defined?

c Were the concept changes triggered internally by members of the development team or externally by the teaching team or representatives of the cooperating industrial company?

d In how many projects was a CFP prototype developed in the CFP phase actually adopted in the funky system prototype?

3.3.2 Methodology and sample of the empirical analyses

Our empirical study investigates students' quarterly team project reports, which document the development process in detail. Since our work focuses on the development of physical products, we pre-selected projects based on the object of development. For this purpose, 124 development projects from the years 2006–2019 were initially classified with regard to the development object and divided into three categories: physical products, software applications and service/business process models. From the 55 projects that aimed to develop physical products, we finally selected 10. To reach this selection, we focused on the time period between 2014 and 2019 and took care to ensure that development documentation allowed for the complete traceability of all development paths. Also, in order to limit company-specific influences on the analysis, we confirmed that no industry sponsor was represented more than once in our final selection. We have included the following development projects in our empirical analysis; with each labelled by the name of the industry sponsor and the year of project completion, they are: VolvoCE (2014), Mabe (2014), Ford (2016), ShoeInn (2016), Renault (2016), IKEA (2016), Audi (2017), Safran (2018), Panasonic (2019), and Volkswagen (2019). The empirical evaluation of their respective development processes thus comprises thirty project reports with a total of 3578 pages. From these project reports, we assessed the technical concept of each prototype and its underlying problem formulation. This allowed us to trace the prototype paths in the projects and to analyse the connections between consecutive prototypes.

Problem formulations, depicting the understanding of the problem, were determined based on verbal descriptions in project reports. The following text passages from the final project report of the VW 2019 project each document different understandings of the underlying problem. Based on the project prompt given by the cooperating industrial company, the development team defines the initial problem formulation:

> "The exact prompt given to us by our corporate partner, VW, is the following: "How will future robotic, on-demand vehicles gracefully and safely accommodate wheelchairs and similar devices?" This prompt is essentially asking us to design a wheelchair securement system that riders can operate independently and is also crash-worthy and ergonomic."(Aravindan et al., 2019, p.23)

The initial problem formulation is then being changed before DHPs are built:

> "Although our team's problem statement is focused on the securement of the wheelchair, our team notices that throughout our testing from Fall quarter, a common note from users was that it is very difficult for them to manoeuvre their wheelchairs in tight spaces. […] For our team, a prototype focused on something other than the securement of the wheelchair would be a dark prototype because it is tackling a different problem space than the one, we had been focused on all of Fall quarter."(Aravindan et al., 2019, pp.157–158)

Finally, based on the knowledge gained during the testing of the FCP, the final problem formulation is defined, which is then extended to the problem of wheelchair users boarding the autonomous taxi:

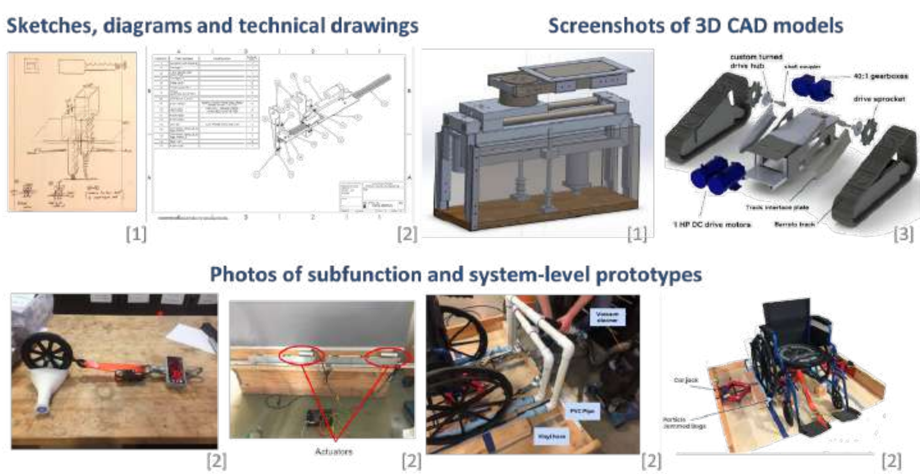
> "Immediately following our return from visiting UNAM over Spring break, we knew we wanted to focus on the issue of entry, and to do that we wanted to prototype different ways of automating the ramp." (Aravindan et al., 2019, p.81)

The analysis of the prototypes' technical concept, which is defined by the effective principles of the functional carriers and their connection to fulfil the prototype's solution-determining main functions, was based on two pillars:

1 the verbal descriptions in the project reports

2 the evaluation of the development artefacts depicted in the project reports, such as sketches, diagrams, technical drawings, screenshots of 3D CAD models, and photos of the built prototypes (Figure 15).

Figure 15 Examples of illustrated development artefacts (see online version for colours)



Source: Pictures taken from Al-Khalil et al. (2016), Aravindan et al. (2019) and Brody et al. (2014))

The following passage, which also comes from the report on the VW 2019 project, shows an example of how the effective principle of a CFP is described:

> "One of the most interesting ideas that came up during our brainstorming sessions for a CFP was that of a liquid floor, that deforms around any kind of wheelchair or even other objects like suitcases, strollers etc., and then solidifies to lock the object in. Even though it seemed really funky and futuristic at the time, serendipitously, we came across the method of Particle Jamming, which was very much in line with our idea of a liquid floor.
>
> Particle Jamming is a method that utilises the particulate nature of various kinds of grains that allows them to be deformable when air is allowed flow in between them and locks them together when the air is vacuumed out."(Aravindan et al., 2019, p.117)

The next passage describes how this effective principle is dismissed and replaced by a different one for the construction of the FCP:

> "We did away with using particle jamming as a mechanism to clamp the wheels as the clamping force it offered was not comparable to the cost and space that a complex pneumatic system would take."(Aravindan et al., 2019, p.179)

We analysed the verbal descriptions in the project reports, as well as illustrations and photos of experimental setups and tests performed, to determine

1 the objectives associated with building and testing each prototype

2 whether prototypes were developed iteratively or in parallel

3 whether the prototype was tested with external users.

The following excerpt documents the parallel development and testing of several CFPs to determine the most suitable particles for the implementation of the effective principle '*particle jamming*' described above. This evaluation of different geometric and material characteristics of an effective principle corresponds to the objective '*technically verify*' (see Section 3.2). The images referred to in this extract (F.22 and F.23) are shown in Figures 16 and 17.

> "To choose the particles, we tested with three different particles. Coffee grounds, plastic packing beads, and bean bag foam. As a control, they were all tested inside Ziploc vacuum bags (See Figure F.22) To quantifiably compare between the various particles, we used a pull test on one of the wheels of a wheelchair, using ratchet straps and a force scale to measure the force required to dislodge the wheel from the vacuumed bag of particles. The setup we used is shown in Figure F.23.
>
> From these tests, we observed that the bean bag packing foam did not give any significant clamping force. The forces required to dislodge the wheel from the coffee grounds bag and the plastic beads bag were comparable. Since coffee grounds were more readily available and cheaper, we decided to proceed with coffee grounds."(Aravindan et al., 2019, pp.170–171)

Figure 16 Figure F.22 from Aravindan et al. (2019) (see online version for colours)

Figure F.22: Ziploc vacuum bags filled with (from left to right) coffee grounds, bean bag foam, plastic packing beads.

Figure 17 Figure F.23 from Aravindan et al. (2019) (see online version for colours)

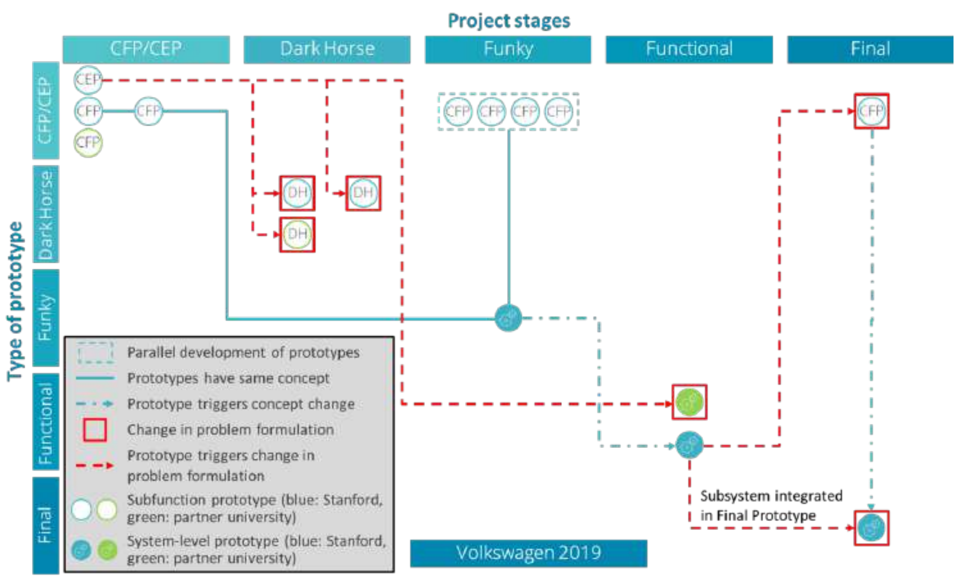
Figure F.23: Setup for testing the force required to pull the wheel of a wheelchair from particle jammed bags filled with different particles

The entire selection process and analysis outlined above was conducted independently by two of our senior researchers specialising in industrial product development. If differences in their assessments arose, their results were consolidated with yet another expert's assistance.

3.3.3 Results and discussion

Figure 18 shows an example of a project's prototype paths, depicting connections between prototypes, concept changes, and changes to the underlying problem formulation. The figure also illustrates the level of detail involved in our analysis of individual projects' development paths (for complete database and supporting information regarding possible connections between prototypes, see Appendices A–C). The results of the research questions formulated in Section 3.3.1 are presented and discussed below.

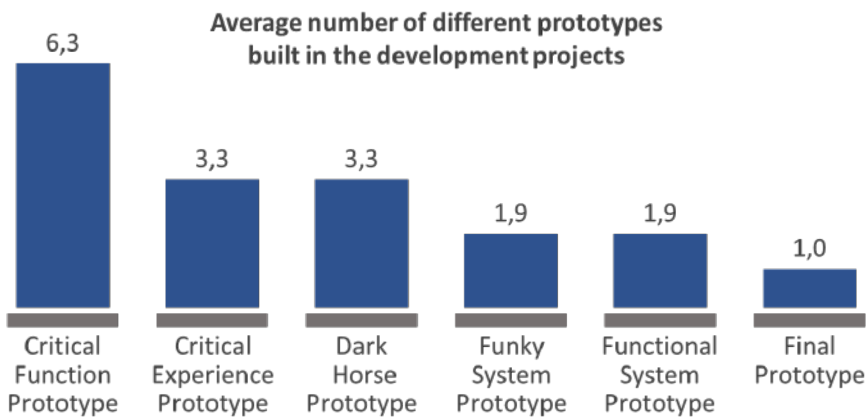
Figure 18 Visualisation of the prototype paths of the VW 2019 project (see online version for colours)



Research question 1: How many of the different prototypes were built on average in the development projects?

Figure 19 shows the average number of different kinds of prototypes built in the 10 projects examined. Critical function prototypes were built most frequently with an average of 6.3 CFPs per project. Considering that CFPs are the main instrument for developing and validating effective principles for solution-determining subfunctions in the ME310 process model, this number seems small and indicates selective, point-based exploration of the solution space even at the subsystem level. The same applies for CEPs, which were built second most frequently at 3.3 per project. Given that CEPs are intended to explore the problem space and guide the direction of development, this number also seems rather low. In comparison to that, the number of DHPs, which were built on average as frequently as CEPs, seems high. This indicates that particular emphasis was placed on the DHP phase of the project, which involves building radically new and unconventional solutions. On the system level, only 1.9 FKPs and FCPs were built on average in the projects. The small numbers of system prototypes built illustrates that – as described in Section 2.2 – agile solution development at the overall system level is hardly possible for complex physical products. In each project, one Final Prototype was eventually built, marking the end of development.

Figure 19 Results for research question 1 (see online version for colours)

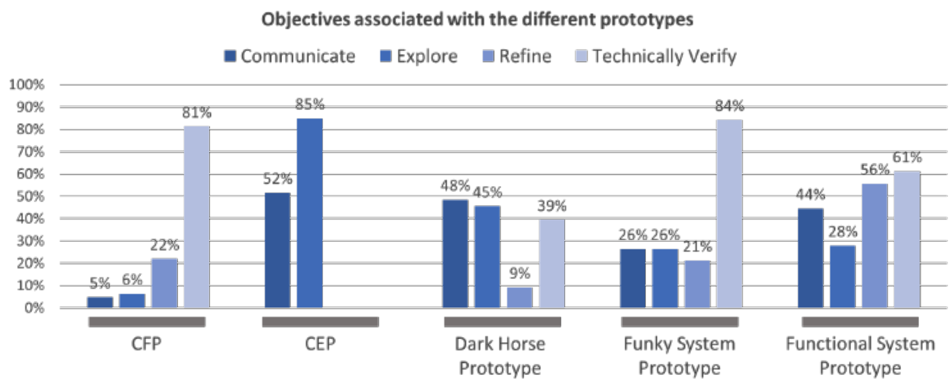


Research question 2: Which objectives are associated with the different prototypes?

Figure 20 shows the goals associated with building and testing the different prototypes in the projects. According to Section 3.2, the goals were divided into explore, communicate, technically verify and refine. Although several objectives were usually associated with the construction and testing of a prototype, distinct objective profiles can be identified for the different types of prototypes that conform to the objectives intended for the respective prototype by the process model. The comparison of the subfunction prototypes CFP and CEP showed that the focus of the CFPs, at 81%, was on the technical verification of effective principles, while the CEPs, at 85%, were mainly used for exploring the problem space. At 52%, the CEPs were also frequently used to communicate ideas. Both the 'communicate' and 'explore' objective are closely linked since identifying user needs in user interaction tests also requires communicating the idea and the direction of development associated with it. In the case of the CFPs, the refinement of the solution was, at 22%, the second most important objective, since the built CFPs were often iteratively further developed in order to optimise the material and geometric characteristics of the respective effective principle.

The DHPs were used almost equally for the objectives 'communicate' (48%), 'explore' (45%) and 'technically verify' (39%). This seems plausible in light of the general purpose of the DHP according to the ME310 process model: The DHPs should critically question the underlying understanding of the problem and the basic direction of development associated with it ("explore") and, at the same time, verify the suitability of the often-risky effective principles ("technically verify"). Since the solutions embodied for this purpose are usually unconventional, the communication of these solutions is also of particular importance.

Figure 20 Results for research question 2 (see online version for colours)

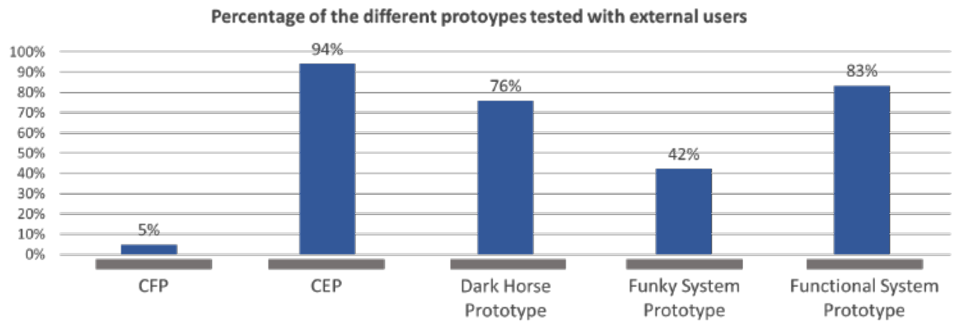


With the FKP, which is the first prototype to realise an effective concept at the overall system level, the focus, at 84%, was clearly on the technical verification of this concept. The FCP, on the other hand, showed a more balanced picture: The two objectives, 'refine' and 'technically verify', were almost equally important here, with 56% and 61%, respectively. The importance of the objective 'refine' is due to the fact that the FCP should already have a perceived quality comparable to the later product, i.e., the resolution of the FCP has to be greatly increased compared to the FKP. The high relevance of the objective 'technically verify' results from the increase in the functional scope, i.e., the larger number of function carriers and the associated increased technical complexity of the prototype, which must be physically assured accordingly.

Research question 3: What percentage of the different prototypes were tested with external users?

Figure 21 shows the percentage of prototypes that were tested with the help of external users. The results are consistent with the objectives intended for the respective prototypes according to the process model. While only a few CFPs (5%) were tested involving external users, this is true for almost all CEPs (94%). This also applies to the DHPs and the system prototypes. The DHPs were tested 76% of the time with external users. Critically reviewing the underlying understanding of the problem using the DHP requires a high degree of user integration. Since the focus of the FKPs is primarily on the technical verification of the overall concept, only 42% were tested with external users. In terms of usability and design, the FCP should already have a value proposition comparable to the final prototype. In contrast to the FKP, this requires much greater involvement of external users, which is reflected in the value of 83%. Overall, the results illustrate the intensive involvement of potential users in the development process, thus underscoring the user-centeredness of the development approach.

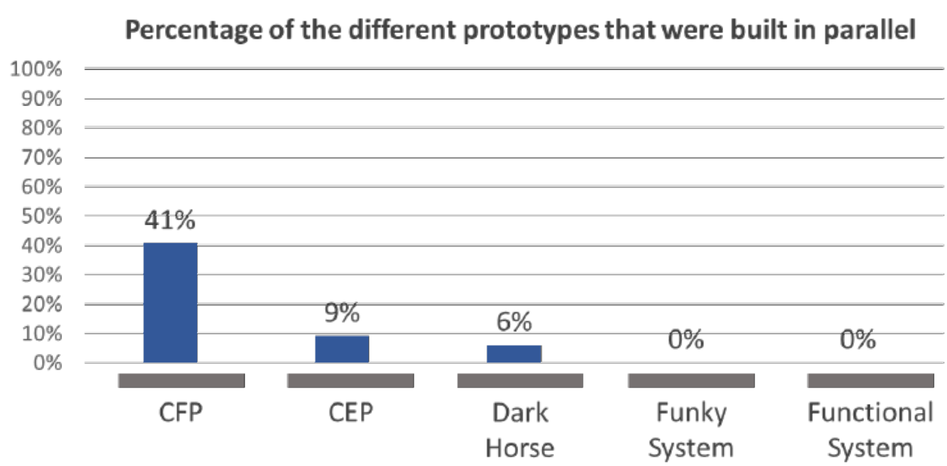
Figure 21 Results for research question 3 (see online version for colours)



Research question 4: What percentage of the different prototypes were built in parallel?

Figure 22 shows the extent to which parallel prototyping took place for the different kinds of prototypes. It is evident that parallel prototyping only took place for the subfunction prototypes and here only for the CFPs to any significant extent. For the system prototypes, the development entirely followed an iterative point-based approach. This result reflects the fundamental problem associated with a prototype-based development approach explained in Section 2.2.

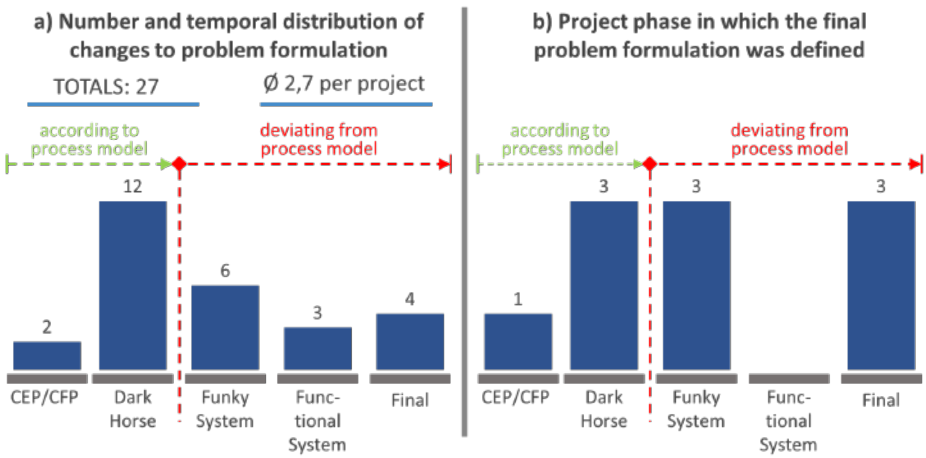
Figure 22 Results for research question 4 (see online version for colours)



Research question 5: Does the development of the problem formulation follow the theoretical specifications of the process model?

Figure 23 shows how often and in which project phases problem formulations changed and when the final problem formulation was established. In the CEP/CFP phase, during which the actual coevolution of problem and solution space should take place according to the ME310 process model, an adjustment of the underlying problem formulation rarely occurred. Most changes to the problem formulation took place during the Dark Horse phase, which, corresponding to the process model, is a phase intentionally devoted to the critical questioning of the previously developed problem formulation. The problem formulation was, however, still frequently modified in the subsequent system prototype phases, in particular the FKP phase, despite the specifications of the ME310 process model. 63% of changes to the underlying problem formulation were triggered internally within the student teams. In only four of 10 projects we examined, the final problem formulation was determined (as specified by the process model) upon completion of the Dark Horse phase. However, in three projects, it was determined no earlier than the Final Prototype phase.

Figure 23 Results for research question 5(a) and (b) (see online version for colours)

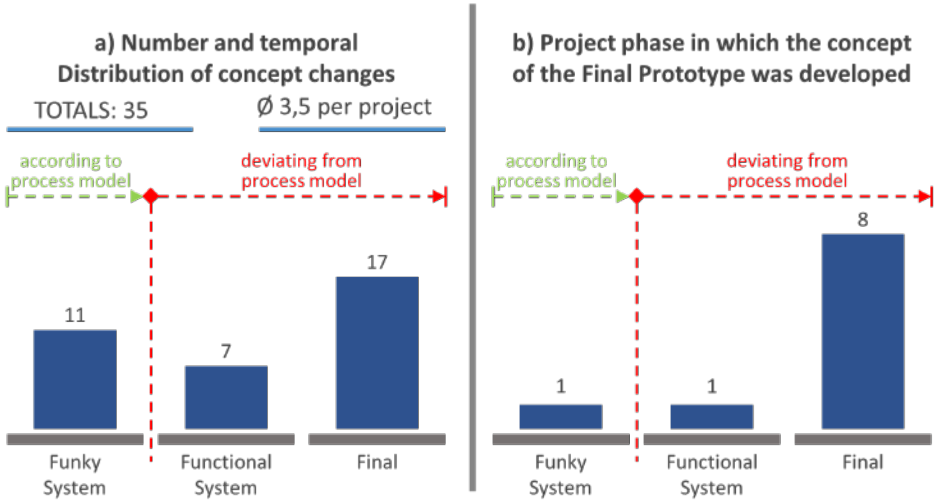


A possible explanation for the frequent reframing of the problem in the system prototype phases is that each prototype built in the CEP/CFP phase only represents the effective principle or user experience of different subfunctions. Thus, these prototypes do not provide a sufficient basis for questions that can contribute to the development of a comprehensive understanding of the problem on the overall system level. This could also explain the relatively high number of problem space changes in the FKP Phase, in which, for the first time, a prototype is built that roughly embodies the solution-determining main functions and their interaction on the overall system level.

Research question 6: Does the concept development follow the theoretical specifications of the process model?

Figure 24 shows, starting with the FKP phase, when and how often the technical concept was changed and the phase in which the concept of the Final Prototype was developed. The concept initially defined in the FKP phase is changed frequently during the very same phase. Even after completion of the FKP phase, one can observe a high volatility of the technical concepts in play. Remarkably, even in the Final Prototype phase, there were still 17 concept changes spread over eight development projects. 78% of these concept changes were triggered within the team. Only in one project was the final concept, conforming to the specifications of the ME310 process model, determined after completion of the FKP phase. In eight projects, however, the concept of the Final Prototype was only defined in the last development phase. Furthermore, only in one out of 10 projects was a function carrier, respectively an effective principle, from the CFP phase part of the FKP. In all projects investigated, CEPs and/or CFPs are built up again in the later system prototype phases.

Figure 24 Results for research question 6(a) and (b) (see online version for colours)



As expected, the changes in the problem space, which often required at least a partial restart in development, also entailed changes in the technical solution concept. Concept development showed an overall higher and prolonged level of volatility compared to the problem formulation. Thus, on average, the concept was changed 1.8 times after the final problem formulation was defined.

One reason for the low number of function carriers transferred from the CFP phase to the FKP may be the lack of methodological support for the design of the overall system. Since an analysis of the effective interrelationships between the function carriers is missing in the ME310 process model, evaluating whether they can be reasonably combined to an overall solution is considerably more difficult. The frequent concept changes at the overall system level can probably be attributed to the stronger point-based orientation associated with the prototype-based development approach. In contrast to systems engineering, there is no parallel development and evaluation of different concepts so that the explored solution space on the overall system level is comparably smaller at the time of concept decision, which, consequently, is detrimental to the stability of the concept decision.

3.4 Conclusion from the empirical and theoretical analysis and derivation of specifications for the design of a new hybrid process model

Reflecting the findings from the theoretical analysis of the ME310 process model, the results of our empirical analysis show that the development approach is characterised by intensive user-centeredness. The defined prototypes fulfil their intended purposes in the practical implementation of the design methodology. However, it also turned out that the observed development processes did not follow the convergence path specified by the process model. Even after completion of the Dark Horse phase, changes in the underlying understanding of the problem occurred frequently. The same applies to the technical concept, which was also still frequently changed after completion of the FKP phase and, in most projects, was ultimately defined no earlier than in the Final Prototype phase. On the one hand, this deviation from the theoretical convergence path of the process model can be attributed to the insufficient methodological support of the overall system development. Without a functional modelling of the overall system, the basis for the explicit functional decomposition of the overall system and the analysis of the functional relationships between function carriers in the overall system is missing. This makes it considerably more difficult to identify critical functions and to determine reasonable and compatible combinations of function carriers. These issues first became the focus of interest within the FKP phase and often led to a complete reorientation in both the problem and solution space in the projects. This is impressively shown in the frequent concept changes in the FKP phase and in the low rate of CFPs transferred from the CFP/CEP phase to the FKP. On the other hand, there is a lack of theoretical exploration of a larger solution space prior to the concept decision. The small number of prototypes built and the low degree of parallel development illustrate that the solution space was explored rather selectively, which has a negative impact on the stability of the concept decision.

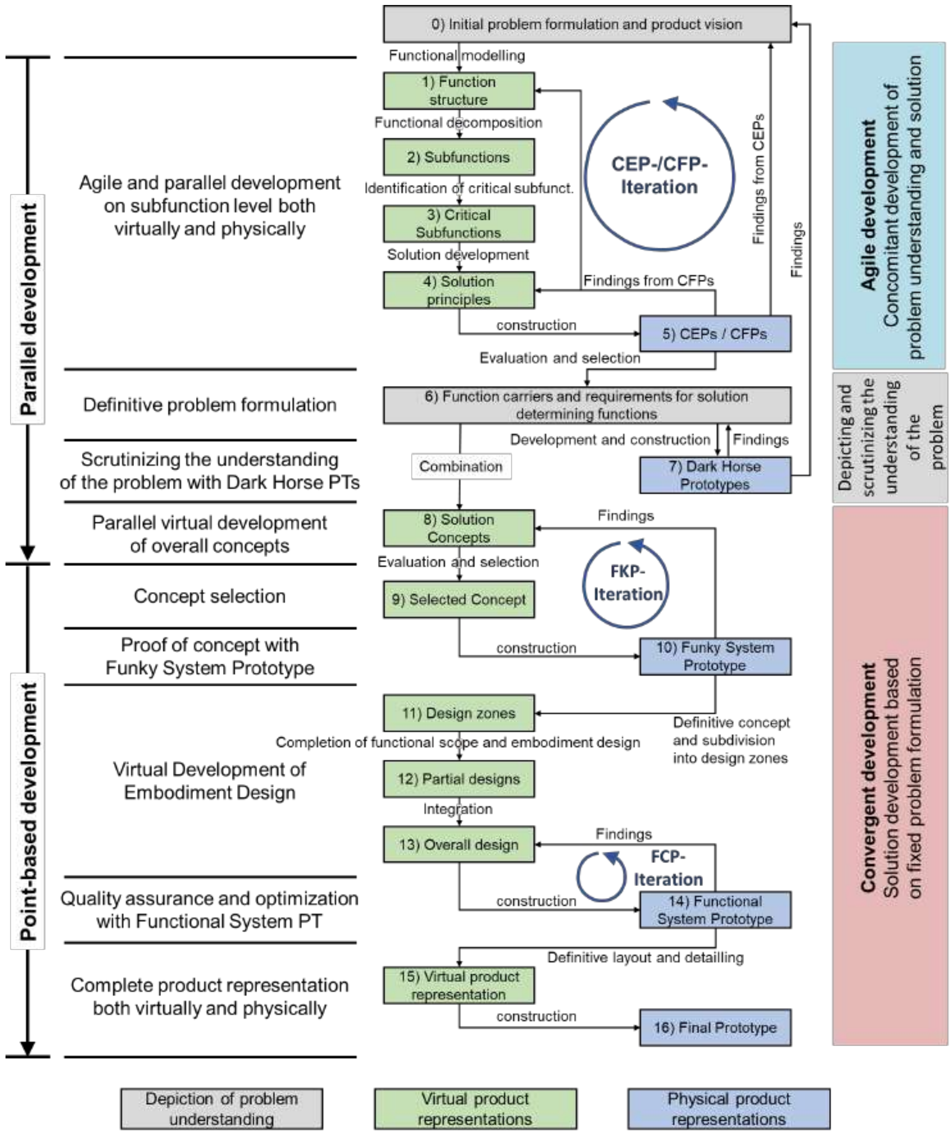
For the development of a new hybrid process model based on the ME310 process model, the following three specifications can therefore be derived:

1. Before starting the search for solutions at the subsystem level, a functional modelling of the overall system with explicit functional decomposition must be carried out.
2. The prototype-based solution development must be supplemented by a virtual development in the early phases of the development process in order to enable the theoretical exploration of a larger solution space.
3. Prior to the concept decision a virtual parallel development must at least take place at the overall system level as well, in which several competing concepts are developed and evaluated.

4 Systematic engineering-design-thinking (SEDT) – a new process model for the hybrid development of physical products

Figure 25 shows our newly developed hybrid product development process model: systematic engineering-design-thinking (SEDT), which builds on the ME310 process model. The needfinding and benchmarking phases, which are used in the ME310 process model to develop an initial understanding of the problem and a product vision before the actual solution development begins, remain unchanged. Our process model builds on the ME310 process model´s macro-logic, meaning that the different types of prototypes from the ME310 process model also serve as milestones to structure the development process and guide the convergence of solution development in our process model. Thus, the basic structure of the ME310 process model with an agile development on the subsystem level, the subsequent formulation and validation of the problem understanding and finally a convergent development on the overall system level, has also been adopted.

Figure 25 Systematic engineering-design-thinking (SEDT) (see online version for colours)



Our changes focus on the phases of actual solution development, which have been extended and supplemented at several points in order to implement the specifications derived in the previous section. For this purpose, selected methods from systems engineering were integrated into the ME310 process model to enable the theoretical exploration of a larger solution space and to provide methodological support for the overall system design. The individual sections of the new process model are described below.

Step 1–5: The CEP-/CFP-iteration – agile development of principle solution at the subsystem level

Before building the CEPs and CFPs, a function structure for the overall system is established, which is used for identifying the solution-determining subfunctions. For said subfunctions, CEPs and CFPs are built and tested accordingly. The findings from testing the CFPs are used to correct and detail the functional modelling of the system and guide the development and selection of effective principles. The CEPs, on the other hand, deepen the understanding of the underlying problem and can sometimes even lead to a reorientation with respect to the initial problem understanding and the product vision based on it. In this phase, the process model is completely agile.

Step 6–7: Requirements and Dark Horse prototype – depicting and scrutinising the problem

The depiction and scrutinisation of the problem understanding corresponds to the procedure in the ME310 process model. After completion of solution development at the subsystem level, first, the underlying problem understanding is depicted by formulating functional and physical requirements. With the construction of DHPs, this problem understanding is then critically scrutinised. As a result, the DHPs can lead to both a confirmation and a change of the underlying problem understanding. In the latter case, a return to the CEP/CFP phase takes place.

Step 8–10: Funky system prototype iteration – concept development at the overall system level

As in the ME310 process model, concept development, i.e., the development of a principle solution at the overall system level, builds on a fixed and validated understanding of the problem, meaning it is not agile, but aimed at convergence. The difference to the ME310 process model is that the construction of the FKP is preceded by a parallel virtual concept development. Here, just as in systems engineering, multiple concepts are developed by combining different physically and geometrically compatible solutions at the subsystem level into effective structures at the overall system level. This procedure can be supported by using a morphological chart. The evaluation of the developed concepts can build on the knowledge gained during the practical testing of CEPs and CFPs. The technical verification of solution-determining function carriers, in combination with functional modeling, leads to an improved understanding of the interaction of these function carriers in the overall system, increasing the reliability of concept evaluation and therefore the stability of concept selection. The FKP's main purpose is physical concept verification. Since the process model is no longer agile at this stage, an iterative design or testing of several different FKPs is not intended. Nevertheless, unanticipated technical problems might become evident during construction or testing of the FKP, which might force a modification or even fundamental change of the technical concept leading to at least a partial redesign of the FKP.

Step 11–14: Functional system prototype iteration – embodiment design

The process of developing the embodiment design after concept selection and verification based on the FKP is similar to the VDI 2221 (2019) and Pahl et al. (2007). First, the concept is subdivided into modules representing the embodiment-determining main function carriers. As in systems engineering, actual embodiment design then starts at the subsystem level by creating partial designs for the defined modules which are eventually integrated into an overall design physically represented by the FCP. During this process the functional scope, which in the FKP was still limited to the solution-determining flow of the function structure, is completed and formal aesthetic aspects are also considered. Since the FCP represents the complete functional scope of the later product, its design and construction are correspondingly time-consuming. At this point, the solution development should therefore have already converged to such an extent that the iteration based on the FCP is limited to the optimisation of partial aspects of the solution. Such an optimisation can, for example, refer to the improvement of certain subfunctions, the interaction of system components or the quality impression. A complete redesign of solution-determining function carriers, or even of the entire FCP, is explicitly not intended within this process model.

Step 15–16: The final prototype – completion of development

In the final development phase, the virtual product representation is detailed and, if necessary, corrected based on the findings obtained during the testing of the FCP. The manufacturing plan and the final documentation are elaborated upon. Lastly, the Final Prototype is built, representing the complete user experience of a product to be industrially realised and marking the completion of development.

5 Conclusion and outlook

In this paper, we have designed a new hybrid process model for the development of physical products based on systems engineering and design thinking. For this purpose, we first defined the fundamental principle of agile development with respect to engineering design methodology and illustrated the implications of the agile development of physical products. Subsequently, we analysed Stanford´s hybrid ME310 process model both theoretically and empirically to investigate its practical suitability for the development of physical products. Our analysis has revealed deficits in Stanford's process model with respect to solution space exploration that impeded convergence in solution development. From the identified deficits, we have derived specifications for the design of a new hybrid process model. Said specifications have been implemented accordingly in the development of the process model SEDT. SEDT combines the ME310 process model´s agile design thinking approach with the systematic solution space exploration of systems engineering to reap the benefits of both design methodologies in the development of physical products.

Further research should aim at validating the newly developed process model by proving its practical suitability in real development processes. This can initially be done in the controlled academic environment of student development projects at the graduate level. Based on the knowledge gained during these investigations, the process model can be further developed iteratively, if necessary, before it is first applied in real world industrial product development. A general difficulty here is the problem of developing and operationalising criteria for evaluating the suitability of a development process model, as described in Section 3.1. The longer time horizon of industrial development processes and the large number of actors involved also considerably increase the effort required for process tracking. Furthermore, there are many internal and external factors that also affect the development process, making it difficult to determine the specific influence of the process model.

In addition, the development project organisations of most manufacturing companies are still strongly oriented towards the process models of systems engineering thus including a strict separation between the responsibility for developing the understanding of the problem and the responsibility for developing the solution. Before using the newly developed process model in industrial practice, a project organisation must first be established that integrates this responsibility organisationally and thus supports the coevolution of problem and solution space in the early phase of the development process.

Funding

This research received no specific grant from any funding agency, commercial or not-for-profit sectors.

Competing interests

The authors declare none.

References

Al-Khalil, H., Foba Amon Junior, A., Wang, C., Slåttsveen, K.B., Solvang, F., Winjum, J. and Hjetland S. (2016) *Enhancing the User Experience of Autonomous Vehicles. Mechanical Engineering* 310 *Spring Design Document*, Final Project Report, Renault 2016, ME310 Project Based Engineering Design, Available at. https://stacks.stanford.edu/file/druid:gw697cc5138/Renault_SpringDoc.pdf (Accessed 27 August, 2022).

Aravindan, A., Wenzel, C.J., Da Jung, H., Varshney, A., Luna, C., Hernandez, S., Lopez, E. and Martinez, I. (2019) *Team VW Final Design Documentation*, Final Project Report, VW. (2019) *ME310 Project Based Engineering Design*, Available at. https://stacks.stanford.edu/file/druid:zp.684gc4352/VW_Spring_FinalDocumentation.pdf (Accessed 19 August, 2022)

Atzberger, A., Nicklas, S.J., Schrof, J., Weiss, S. and Paetzold, K. (2020) *Agile Entwicklung physischer Produkte: Eine Studie zum aktuellen Stand in der industriellen Praxis*, München, Neubiberg, Deutschland, https://doi.org/10.18726/2020_5

Badke-Schaub, P., Roozenburg, N. and Cardoso, C. (2010) 'Design thinking: a paradigm on its way from dilution to meaninglessness?', in Dorst, K., Stewart, S., Staudinger, I., Paton, B. and Dong, A. (Eds.): *Proceedings of the 8th Design Thinking Research Symposium* (*DTRS8*), Sydney, Australia, 19–20 October, DAB documents, pp.39–49.

Bagno, R.B., Salerno, M.S. and Da Silva, D.O. (2017) 'Models with graphical representation for innovation management: a literature review', *R & D Management*, Vol. 47, No. 4, pp.637–653, https://doi.org/10.1111/radm.12254

Berglund, A. and Leifer, L. (2013) 'Why we prototype! an international comparison of the linkage between embedded knowledge and objective learning', *Engineering Education*, Vol. 8, No. 1, pp.2–15, https://doi.org/10.11120/ened.2013.00004

Böhmer, A.I., Hostettler, R., Richter, C., Lindemann, U., Conradt, J. and Knoll, A. (2017) 'Towards agile product development – the role of prototyping', *Proceedings of the 21st International Conference on Engineering Design (ICED17): Vol. 4: Design Methods and Tools*, 21–25 August, Vancouver, Canada, pp.1–10.

Brereton, M. and McGarry, B. (2000) 'An observational study of how objects support engineering design thinking and communication: implications for the design of tangible media', in Turner, T. and Szwillus, G. (Eds.): *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1–6 April, The Hague, The Netherlands, pp.217–224, https://doi.org/10.1145/332040.332434

Brody, J., Martin, T., Hughes, C., Alfred, K., Andersson, D., Chai, Y. and Gao, Z. (2014) *Volvo Construction Equipment 310X Final Documentation 2013-204*, Final Project Report, VolvoCE 2014, ME310 Project Based Engineering Design, Available at. https://stacks.stanford.edu/file/druid:td675km9733/VolvoCE_documentation_Final.pdf (Accessed 27 August, 2022),

Bushnell, T., Steber, S., Matta, A., Cutkosky, M. and Leifer, L. (2013) 'Using a 'Dark Horse' prototype to manage innovative teams', in Fernandes, A.A., Natal Jorge, R.M., Patrício, L. and Medeiros, A. (Eds.): *3rd Int. Conf. on Integration of Design, Engineering and Management for Innovation*, 4–6 September, Polo, Portugal, https://doi.org/10.13140/2.1.2361.7602

Camburn, B., Viswanathan, V., Linsey, J., Anderson, D., Jensen, D., Crawford, R., Otto, K. and Wood, K. (2017) 'Design prototyping methods: state of the art in strategies, techniques, and guidelines', *Design Science*, Vol. 3, No. 13, pp.1–33, https://doi.org/10.1017/dsj.2017.10

Carleton, T. (Ed.): (2019) *ME310 at Stanford University: 50 years of redesign (1967-2017)*, Innovation Leadership Publishing, California.

Carleton, T. and Leifer, L. (2009) 'Stanford´s ME310 course as an evolution of engineering design', *Proceedings of the 19th CIRP Design Conference – Competitive Design*, 30–31 March, Cranfield University, Cranfield, UK, pp.547–554.

Carlgren, L., Elmqvist, M. and Rauth, I. (2014) 'Exploring the use of design thinking in large organizations: towards a research agenda', *Swedish Design Research Journal*, Vol. 11, No. 1, pp.23–32.

Carlgren, L., Rauth, I. and Elmquist, M. (2016) 'Framing design thinking: the concept in idea and enactment', *Creativity and Innovation Management*, Vol. 25, No. 1, pp.38–57, https://doi.org/10.1111/caim.12153

Cocchi, N., Dosi, C. and Vignoli, M. (2021) 'The hybrid model matrix. enhancing stage-gate with design thinking, lean startup, and agile', *Research-Technology Management*, Vol. 64, No. 5, pp.18–30, https://doi.org/10.1080/08956308.2021.1942645

Cocchi, N., Dosi, C. and Vignoli, M. (2024) 'Stage-gate hybridization beyond agile: conceptual review, synthesis, and research agenda', *IEEE Transactions on Engineering Management*, Vol. 71, pp.6435–6453, https://doi.org/10.1109/TEM.2023.3282269

Conforto, E.C., Salum, F., Amaral, D.C., Da Silva, S.L. and de Almeida, L.F.M. (2014) 'Can agile project management be adopted by industries other than software development?', *Project Management Journal*, Vol. 45, No. 3, pp.21–34, https://doi.org/10.1002/pmj.21410

Cooper, R.G. and Sommer, A.F. (2016a) 'Agile-stage-gate: New idea-to-launch method for manufactured new products is faster, more responsive', *Industrial Marketing Management*, Vol. 59, pp.167–180, https://doi.org/10.1016/j.indmarman.2016.10.006

Cooper, R.G. and Sommer, A.F. (2016b) 'The agile-stage-gate hybrid model: a promising new approach and a new research opportunity', *Journal of Product Innovation Management*, Vol. 33, No. 5, pp.513–526, https://doi.org/10.1111/jpim.12314

Cooper, R.G. and Sommer, A.F. (2018) 'Agile–stage-gate for manufacturers', *Research-Technology Management*, Vol. 61, No. 2, pp.17–26, https://doi.org/10.1080/08956308.2018.1421380

Denning, S. (2016) 'How to make the whole organization "Agile"', *Strategy and Leadership*, Vol. 44, No. 4, pp.10–17, https://doi.org/10.1108/SL-06-2016-0043

Domingo, L., Moore, D., Sirkin, D., Toye, G., Leifer, L. and Cutkosky, M. (2020) 'Strategic prototyping to learn in Stanford University´s ME310 design innovation course', in Marjanović, D., Storga, M., Pavković, N. and Bojčetić, N. (Eds.): *DS 102: Proceedings of the DESIGN 2020 16th International Design Conference*, held online, 26–29 October, Cambridge University Press, Cambridge, UK, pp.1687–1696, https://doi.org/10.1017/dsd.2020.176

Dorst, K. and Cross, N. (2001) 'Creativity in the design process: co-evolution of problem–solution', *Design Studies*, Vol. 22, No. 5, pp.425–437, https://doi.org/10.1016/S0142-694X(01)00009-6

Dow, S.P., Glassco, A., Kass, J., Schwarz, M., Schwartz, D.L. and Klemmer, S.R. (2010) 'Parallel prototyping leads to better design results, more divergence, and increased self-efficacy', *ACM Transactions on Computer-Human Interaction*, Vol. 17, No. 4, pp.1–24, https://doi.org/10.1145/1879831.1879836

Edelman, J.A., Leifer, L., Banerjee, B., Jung, M., Sonalkar, N. and Lande, M. (2009) 'Hidden in plain sight: affordances of shared models in team based design', *Proceedings of the 17th International Conference on Engineering Design* (*ICED09*), *Vol. 2, Design Theory and Research Methodology*, 24–27 August, Palo Alto, CA, USA, pp.395–406.

Edwards, K., Cooper, R.G., Vedsmand T. and Nardelli, G. (2019) 'Evaluating the agile-stage-gate hybrid model: experiences from three SME manufacturing firms', *International Journal of Innovation and Technology Management*, Vol. 16, No. 08, p.1950048, https://doi.org/10.1142/S0219877019500482

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

Gericke, K., Beinke, C., Gemmer, P. and Blessing, L. (2010) 'Entwicklungsmethodik nach Pahl Beitz und Design Thinking. Vergleich und Einordnung', in Krause, D., Paetzold, K. and Wartzack, S. (Eds.): *DFX 2010: Proceedings of the 21st Symposium on Design for X*, 23–24 September, Buchholz, Hamburg, Deutschland, pp.117–130.

Gerstbach, I. (2017) *77 Tools für Design Thinker: Insidertipps aus der Design-Thinking-Praxis, Whitebooks*, GABAL Verlag, Offenbach.

Goevert, K., Heimicke, J., Lindemann, U. and Albers, A. (2019) 'Interview study on the agile development of mechatronic systems', *Proceedings of the 22nd International Conference on Engineering Design (ICED19)*, 5–8 August, Delft, The Netherlands, pp.2287–2296, https://doi.org/10.1017/dsi.2019.235

Grauvogl, C.J. (2018) *Prototype Centered Development – A Compass to Tangible Results in the Fuzzy Front End of Innovation*, Master's Thesis, Department of Mechanical Engineering, Technical University of Munich, Munich.

Gruber, M., Leon, N.d., George, G. and Thompson, P. (2015) 'Managing by design', *Academy of Management Journal*, Vol. 58, No. 1, pp.1–7, https://doi.org/10.5465/amj.2015.4001

Haberfellner, R., de Weck, O.L., Fricke, E. and Vössner, S. (2019) *Systems Engineering: Fundamentals and Applications*, Springer Nature, Cham, Switzerland.

Hallgrimsson, B. (2012) *Prototyping and Modelmaking for Product Design*, Laurence King, London.

Hassi, L. and Laakso, M. (2011) 'Conceptions of design thinking in the design and management discourses. open questions and possible directions for research', in Roozenburg, N., Chen, L-L. and Stappers, P.J. (Eds.): *Proceedings of the IASDR 2011 Delft Diversity and Unity, the 4th World Conference on Design Research*, 31 October–4 November, Delft, The Netherlands, pp.1–10.

Heimicke, J., Chen, R. and Albers, A. (2020) 'Agile meets plan-driven – hybrid approaches in product development: a systematic literature review', in Marjanović, D., Storga, M., Pavković, N. and Bojčetić, N. (Eds.): *DS 102: Proceedings of the DESIGN 2020 16th International Design Conference*, held online, 26–29 October, Cambridge University Press, Cambridge, UK, pp.577–586, https://doi.org/10.1017/dsd.2020.259

Johansson-Sköldberg, U., Woodilla, J. and Çetinkaya, M. (2013) 'Design thinking: past, present and possible futures', *Creativity and Innovation Management*, Vol. 22, No. 2, pp.121–146, https://doi.org/10.1111/caim.12023

Kenyon, D., Cutkowsky, M. and Leifer, L. (n.d.) *ME310 ABC Course Reader*, Stanford University, Stanford, California, USA.

Kersten, W. and Koppenhagen, F. (2002) 'Systematische Ableitung modularer Produktarchitekturen', *PPS-Management*, Vol. 7, No. 1, pp.9–13.

Kettunen, J., Grushka-Cockayne, Y., Degraeve, Z. and de Reyck, B. (2015) 'New product development flexibility in a competitive environment', *European Journal of Operational Research*, Vol. 244, No. 3, pp.892–904, https://doi.org/10.1016/j.ejor.2015.02.016

Kimbell, L. (2011) 'Rethinking design thinking: part I', *Design and Culture*, Vol. 3, No. 3, pp.285–306, https://doi.org/10.2752/175470811X13071166525216

Klein, T. (2016) *Agiles Engineering im Maschinen- und Anlagenbau*, Herbert Utz Verlag, München.

Koppenhagen, F. (2011) 'Verzahnte Produktentwicklung – Wirksames Komplexitätsmanagement durch die kohärente Gestaltung von Produktarchitektur, Entwicklungsorganisation und Entwicklungsprozess', *Industrie Management*, Vol. 27, No. 5, pp.13–16.

Koppenhagen, F. and Wecht, C.H. (2023) 'Warum das nächste iPhone wieder nicht aus Deutschland kommt: Der Zusammenhang zwischen Entwicklungsmethodik und radikalen Produktinnovationen – Ein Vergleich zwischen Stanford und Deutschland/Why the next iPhone won't come from Germany again: the connection between design methodology and radical product innovations – A comparison between Stanford und Germany', *Konstruktion*, Vol. 75, No. 06, pp.68–74, https://doi.org/10.37544/0720-5953-2023-06-68

Koppenhagen, F., Blümel, T., Held, T. and Wecht, C. (2021a) 'Zeitliche und inhaltliche Konvergenz der Lösungsfindung als zentrale Herausforderung in hybriden Produktentwicklungsprozessen - eine empirische Analyse von Stanfords ME310-Prozessmodell', in Stelzer, R.H. and Krzywinski, J. (Eds.): *Entwerfen Entwickeln Erleben in Produktentwicklung und Design 2021*, 17–18 June, TUDpress, Dresden, pp.137–154, https://doi.org/10.25368/2021.17

Koppenhagen, F., Blümel, T., Held, T., Wecht, C. and Kollmer, P.D. (2021b) 'The challenging combination of agility and convergence in hybrid product development processes: an empirical analysis of Stanford's ME310 process model', *Proceedings of the 23rd International Conference on Engineering Design* (*ICED21)*, 16–20 September, Gothenburg, Sweden, pp.2991–3000, https://doi.org/10.1017/pds.2021.560

Lauff, C., Kotys-Schwartz, D. and Rentschler, M.E. (2017) 'What is a prototype? Emergent roles of prototypes from empirical work in three diverse companies', *Proceedings of the ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference: Volume 7: 29th International Conference on Design Theory and Methodology,* 6–9 August, Cleveland, Ohio, USA, https://doi.org/10.1115/DETC2017-67173

Leifer, L. (2012) '"Rede nicht, zeig´s mir!", Über Design Thinking, Bad Guys, Experiment, Jagd und organisationalen Wandel", ein Interview mit Larry Leifer', *OrganisationsEntwicklung – Zeitschrift für Unternehmensentwicklung und Change Management*, Vol. 31, No. 2, pp.8–13.

Liedtka, J. (2015) 'Perspective: Linking design thinking with innovation outcomes through cognitive bias reduction', *Journal of Product Innovation Management*, Vol. 32, No. 6, pp.925–938.

McKim, R.H. (1980) *Experiences in Visual Thinking*, 2nd ed., PWS Publ., Boston.

Osterwalder, A., Pigneur, Y., Bernarda, G., Smith, A. and Papadakos, P. (2014) *Value Proposition Design: How to Create Products and Services Customers Want. Get Started with*, John Wiley & Sons, Hoboken, New Jersey.

Otto, K.N. and Wood, K.L. (2001) *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River, NJ.

Ovesen, N. (2012) *The Challenges of Becoming Agile. Implementing and Conducting SCRUM in Integrated Product Development*, Dissertation, Department of Architecture, Design, and Media Technology, Aalborg University.

Ovesen, N. and Sommer, A.F. (2015) 'Scrum in the traditional development organization: adapting to the legacy', in Schabacker, M., Gericke, K., Szélig, N. and Vajna, S. (Eds.): *Modelling and Management of Engineering Processes*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp.87–99, https://doi.org/10.1007/978-3-662-44009-4_8

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K-H. (2007) *Engineering Design: A Systematic Approach*, 3rd ed., Springer, London, https://doi.org/10.1007/978-1-84628-319-2

Pedersen, K., Emblemsvåg, J., Bailey, R., Allen, J.K. and Mistree, F. (2000) 'Validating design methods and research: the validation square', *Proceedings of DETC'00: 2000 ASME Design Engineering Technical Conferences*, 10–14 September, Baltimore, Maryland, pp.1–13.

Ponn, J. and Lindemann, U. (2011) *Konzeptentwicklung und Gestaltung technischer Produkte: Systematisch von Anforderungen zu Konzepten und Gestaltlösungen*, 2nd ed., Springer, Berlin.

Reinertsen, D.G. (1997) *Managing the Design Factory: A Product Developer´s Toolkit*, The Free Press, New York, NY.

Rittel, H.W.J. and Webber, M.M. (1973) 'Dilemmas in a general theory of planning', *Policy Sciences*, Vol. 4, No. 2, pp.155–169.

Schmidt, T.S. (2019) *Towards a Method for Agile Development in Mechatronics*, Dissertation, Shaker Verlag, Düren.

Schmidt, T.S., Chahin, A., Kößler, J. and Paetzold, K. (2017) 'Agile development and the constraints of physicality: a network theory-based cause-and-effect analysis', *Proceedings of the 21st International Conference on Engineering Design (ICED17): Vol. 4: Design Methods and Tools*, 21–25 August, Vancouver, Canada, pp.199–208.

Schmidt, T.S., Weiss, S. and Paetzold, K. (2018) *Agile Development of Physical Products: An Empirical Study about Motivations, Potentials and Applicability*, Universität der Bundeswehr München, Neubiberg, Germany.

Schön, D.A. (1983) *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York.

Schröder, S. (2020) *Integration von Lieferanten in agile Entwicklungsprozesse technischer Systeme, Edition Wissenschaft Apprimus*, Band, Vol. 6, 1. Auflage, Apprimus Verlag, Aachen.

Schuh, G. and Dölle, C. (2021) *Sustainable Innovation: Nachhaltig Werte schaffen*, 2nd ed., Springer, Berlin, Heidelberg.

Schüttoff, M., Herrmann, T., Roth, D. and Binz, H. (2019) 'Analyse und Beurteilung der unterschiedlichen Einsatzzwecke und Anwendungsgrenzen von Design Thinking', in Binz, H., Bertsche, B., Bauer, W. and Riedel, O. (Eds.): *Stuttgarter Symposium für Produktentwicklung SSP 2019: Agilität und kognitives Engineering*, 16 May, Stuttgart, pp.193–202.

Simpson, D. and Hinkle, G. (2018) *An Introduction to MAHD Modified Agile for Hardware Development. Applying Agile Principles to Physical Products When Making Changes Is Difficult*, Available at. https://agileforhardware.org/wp-content/uploads/2018/07/An-Intro-to-MAHD-Ebook-7_23_18.pdf (Accessed 18 August, 2022).

Sobek II, D.K., Ward, A.C. and Liker, J.K. (1999) 'Toyota´s principles of set-based concurrent engineering', *Sloan Management Review*, Vol. 40, No. 2, pp.67–83.

Soltani, S., Ramazanpoor, E. and Eslamian, S. (2013) 'A structural equation model of the impact of new product development on competitive advantage', *Engineering Management Research*, Vol. 3, No. 1, pp.99–108, https://doi.org/10.5539/emr.v3n1p.99

Sommer, A.F., Hedegaard, C., Dukovska-Popovska, I. and Steger-Jensen, K. (2015) 'Improved product development performance through agile/Stage-gate hybrids: the next-generation stage-gate process?', *Research-Technology Management*, Vol. 58, No. 1, pp.34–45, https://doi.org/10.5437/08956308X5801236

Ulrich, K.T., Eppinger, S.D. and Yang, M.C. (2020) *Product Design and Development*, 7th ed., Mcgraw-Hill Education, New York, NY.

VDI 2221 (2019) *Design of Technical Products and Systems: Model of Product Design*, Part 1, Beuth Verlag, Berlin.

Viswanathan, V.K. and Linsey, J.S. (2012) 'Physical models and design thinking: a study of functionality, novelty and variety of ideas', *Journal of Mechanical Design*, Vol. 134, No. 9, pp.1–13, https://doi.org/10.1115/1.4007148

Appendix A: links to the project reports of the investigated projects

The student project reports analysed in this publication can be found in the Stanford Digital Repository under the links provided in the table. In most cases, project reports of the fall quarter (and sometimes also the winter quarter) were not archived separately but integrated into the final project report of the spring quarter. However, the authors had all 30 reports of the individual quarters (fall, winter and spring) of the 10 investigated projects available for their analysis.

Project name	*Link*
VolVoCE (2014)	https://searchworks.stanford.edu/view/td675km9733
Mabe (2014)	https://searchworks.stanford.edu/view/gk257xf6317
Ford (2016)	https://searchworks.stanford.edu/view/dj561gb3422
ShoeInn (2016)	https://searchworks.stanford.edu/view/yr390pd3005
Renault (2016)	https://searchworks.stanford.edu/view/gw697cc5138
IKEA (2016)	https://searchworks.stanford.edu/view/sc020qp6018
Audi (2017)	https://searchworks.stanford.edu/view/wj733zq9620
Safran (2018)	https://searchworks.stanford.edu/view/vv534gd3897
Panasonic (2019)	https://searchworks.stanford.edu/view/dy270yj7833
Volkswagen (2019)	https://searchworks.stanford.edu/view/zp684gc4352

All links were last accessed on 29 August 2022.

Appendix B: analysis of all prototypes in the 10 projects investigated

Project (free description)	Project prompt (free description)	Final product (free description)	Quarter (Fall; Winter; Spring)	Development phase (CFP/CEP; Dark Horse; Funky; Functional; Final)	Type of Prototype (Critical Function PT; Critical Experience PT; Dark Horse PT; Funky PT; Functional PT; Final PT)	Name of Prototype (free description)	Built by (Stanford / Partner university)	Functionality (monofunctional / multifunctional)	Development mode (point-based/ parallel)	tested with external users? (yes / no)	Communicate	Explore	Refine	Technically Verify	problem formulation fixed	concept fixed
2019 Volkswagen	How will future Robotic, on-demand vehicles gracefully and safely accommodate wheelchairs and similar devices in 2025?	*RISE - RISE (Robot for Independent Securing and Entering) is an assistant robot for SEDRIC, Volkswagen's autonomous taxi. The robot transports the wheelchair user into the vehicle, turns it in the direction of travel and secures the wheelchair inside the SEDRIC.*	Fall	CFP/CEP	Critical Experience Prototype	SEDRIC Mockup	Stanford	-	point-based	yes		x				
			Fall	CFP/CEP	Critical Function Prototype	Particle Jamming	Stanford	monofunctional	point-based	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Particle Jamming Scale Up	Stanford	monofunctional	point-based	no			x	x		
			Fall	CFP/CEP	Critical Function Prototype	Smart Seat Belt Pressure Sensor	Partner	monofunctional	point based	yes				x		
			Winter	Dark Horse	Dark Horse Prototype	Scooper Door	Stanford	monofunctional	point-based	yes		x				
			Winter	Dark Horse	Dark Horse Prototype	Collapsible Storage for Service Animals	Stanford	monofunctional	point-based	yes		x				
			Winter	Dark Horse	Dark Horse Prototype	Moka	Partner	monofunctional	point-based	yes		x				
			Winter	Funky	Critical Function Prototype	PJ Particles - Coffee Ground	Stanford	monofunctional	parallel	no				x		
			Winter	Funky	Critical Function Prototype	PJ Particles - Plastic Packing Beads	Stanford	monofunctional	parallel	no				x		
			Winter	Funky	Critical Function Prototype	PJ Particles - Bean Bag Foam	Stanford	monofunctional	parallel	no				x		
			Winter	Funky	Critical Function Prototype	Particle Jamming Membrane	Stanford	monofunctional	parallel	no		x				
			Winter	Funky	Funky Prototype	Particle Jammed Clamps	Stanford	multifunctional	point-based	no			x	x		
			Winter	Functional	Functional Prototype	Automated Voice Activated Clamps	Stanford	multifunctional	point-based	yes	x	x		x		
			Winter	Functional	Functional Prototype	Turntable	Partner	multifunctional	point-based	yes				x		
			Spring	Final	Critical Function Prototype	Rail System	Stanford	monofunctional	point-based	yes		x		x	x	
			Spring	Final	Final Prototype	Rise	Stanford	multifunctional	point-based	yes	x			x		x
2019 Panasonic	*Movable, flexible reconfigurable living spaces*	*Koko - Koko is a product that can be used as a normal stroller while walking, but can also be reconfigured into a trailer that can be attached to an electric scooter. The product also includes a backpack that can be attached to the stroller. If the backpack is carried by a person, a baby carrier sling can directly be attached to the backpack.*	Fall	CFP/CEP	Critical Function Prototype	PolyPlug	Stanford	monofunctional	point-based	no				x		
			Fall	CFP/CEP	Critical Function Prototype	PolyPlug V2	Stanford	monofunctional	point-based	no			x	x		
			Fall	CFP/CEP	Critical Experience Prototype	Furnishare	Stanford	-	point-based	yes	x	x				
			Fall	CFP/CEP	Critical Experience Prototype	Daysar	Partner	-	point-based	yes		x				
			Fall	CFP/CEP	Critical Experience Prototype	The Butterfly	Partner	-	point-based	yes	x					
			Fall	CFP/CEP	Critical Experience Prototype	The Panapod	Partner	-	point-based	yes		x				
			Fall	CFP/CEP	Critical Experience Prototype	The Modulife	Partner	-	point-based	yes	x					
			Fall	CFP/CEP	Critical Experience Prototype	The Chemax	Partner	-	point-based	yes	x					
			Winter	Dark Horse	Dark Horse Prototype	Sensing Mechanism - Pressure Sensing	Stanford	monofunctional	parallel	yes				x		
			Winter	Dark Horse	Dark Horse Prototype	Sensing Mechanism - Vision Sensing	Stanford	monofunctional	parallel	yes				x		
			Winter	Dark Horse	Dark Horse Prototype	Childin Danger Alert	Stanford	-	point-based	yes	x	x				
			Winter	Dark Horse	Dark Horse Prototype	Turtle Home	Partner	monofunctional	point-based	no	x					
			Winter	Dark Horse	Dark Horse Prototype	Self sustaining Community	Partner	monofunctional	point-based	no	x					
			Winter	Dark Horse	Dark Horse Prototype	Sleeping & Storage for Nomads	Partner	monofunctional	point-based	no	x					
			Winter	Funky	Critical Experience Prototype	Proportion Considerations	Partner	-	point-based	yes		x				
			Winter	Funky	Funky Prototype	Toi Wrangler	Partner	multifunctional	point-based	yes		x		x		
			Winter	Funky	Funky Prototype	Soft Robot	Stanford	multifunctional	point-based	no				x		
			Winter	Functional	Functional Prototype	Scissor Gate	Stanford	multifunctional	point-based	no	x			x		
			Spring	Final	Critical Function Prototype	E-Scooter Connector It. 1	Stanford	monofunctional	point-based	no		x		x		
			Spring	Final	Critical Function Prototype	E-Scooter Connector Pin	Stanford	monofunctional	point-based	no			x	x		
			Spring	Final	Critical Function Prototype	Snap-Fit Connection PT	Stanford	monofunctional	point-based	no			x			
			Spring	Final	Critical Function Prototype	Penult connecter	Stanford	monofunctional	point-based	no			x			
			Spring	Final	Final Prototype	Koko Mover	Stanford	multifunctional	point-based	yes	x			x	x	x
2018 Safran	*Improve Mobility outside the home for active people with disabilities*	*Ryde & Slyde - the Slyde system fits in the trunk of a large SUV to load a wheelchair or other heavy object without the user having to lift the object themselves. The Ryde system provides electric propulsion for a manual wheelchair by adapting commercially available electric scooters to connect to the wheelchair.*	Fall	CFP/CEP	Critical Experience Prototype	Assistance Signaling Prototype	Stanford	-	point-based	yes	x					
			Fall	CFP/CEP	Critical Experience Prototype	The Useful Wheel Chari	Stanford	-	point-based	yes	x	x				
			Fall	CFP/CEP	Critical Function Prototype	Generator Wheelchair	Stanford	monofunctional	point-based	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Attachment for Banked Slopes	Partner	monofunctional	point-based	no				x		
			Winter	Dark Horse	Dark Horse Prototype	Rack and Pinion Rail System	Stanford	monofunctional	point-based	no	x					
			Winter	Dark Horse	Dark Horse Prototype	Descending Prototype	Stanford	monofunctional	point-based	yes		x		x		
			Winter	Dark Horse	Dark Horse Prototype	Ascending Prototype	Stanford	monofunctional	point-based	yes		x		x		
			Winter	Funky	Funky Prototype	Wheelchair Bike Rack	Stanford	multifunctional	point-based	no				x	x	
			Winter	Funky	Funky Prototype	Wheelchair Bike Rack refined	Stanford	multifunctional	point based	no			x			
			Winter	Funky	Funky Prototype	Wheelchair Bike Rack + Cover	Stanford	multifunctional	point-based	no			x			
			Winter	Functional	Critical Experience Prototype	Lifting Wheelchair into the Car	Stanford	-	point-based	no	x	x				
			Winter	Functional	Functional Prototype	Wheelchair Transit Adapter	Stanford	multifunctional	point-based	yes			x	x		
			Spring	Final	Final Prototype	Ryde & Slyde	Stanford	multifunctional	point-based	yes	x		x	x		x

Project *(free description)*	Project prompt (free description)	Final product (free description)	Quarter *(Fall; Winter; Spring)*	Development phase *(CFP/CEP; Dark Horse; Funky; Functional; Final)*	Type of Prototype *(Critical Function PT; Critical Experience PT; Dark Horse PT; Funky PT; Functional PT; Final PT)*	Name of Prototype (free description)	Built by *(Stanford / Partner university)*	Functionality *(monofunctional / multifunctional)*	Development mode (point-based/ parallel)	tested with external users? *(yes / no)*	Communicate	Explore	Refine	Technically Verify	problem formulation fixed	concept fixed
2017 Audi	*Make a shared car feel like your own car with specific attention to personal belonging storage and transportation*	**Hitch** *- Hitch is a small and lightweight rack that is installed on the bike and allows to reconfigure the bike into a mountable state within under a minute. This allows the bike to be attached to almost any car.*	Fall	CFP/CEP	Critical Experience Prototype	Leaving Belongings in Car	Stanford	-	point-based	yes	X	X				
			Fall	CFP/CEP	Critical Function Prototype	Interior - Cotton Fluff	Stanford	monofunctional	parallel	no				X		
			Fall	CFP/CEP	Critical Function Prototype	Interior - Mesh	Stanford	monofunctional	parallel	no				X		
			Fall	CFP/CEP	Critical Function Prototype	Interior - Bristle	Stanford	monofunctional	parallel	no				X		
			Fall	CFP/CEP	Critical Experience Prototype	Car Handover	Partner	-	point-based	yes		X				
			Fall	CFP/CEP	Critical Function Prototype	Car Handover Speech in and Output	Partner	monofunctional	point-based	no				X		
			Winter	Dark Horse	Dark Horse Prototype	Hitchhiking Robot	Stanford	multifunctional	point-based	yes	X			X		
			Winter	Funky	Funky Prototype	Smart Beacon	Stanford	multifunctional	point-based	yes	X			X		
			Winter	Funky	Critical Experience Prototype	Smart Beacon App	Stanford	-	point-based	yes	X	X				
			Winter	Functional	Functional Prototype	Carrus System	Stanford	multifunctional	point-based	yes	X	X		X		
			Spring	Final	Critical Function Prototype	Magnet Mount	Stanford	monofunctional	point-based	no				X		
			Spring	Final	Critical Function Prototype	Suction Mount	Stanford	monofunctional	point-based	no				X		
			Spring	Final	Critical Function Prototype	Magnet Cable config	Stanford	monofunctional	point-based	no			X	X		
			Spring	Final	Final Prototype	Hitch	Stanford	multifunctional	point-based	yes	X		X	X	X	X
2016 IKEA	*Creating an Opportunity in the Mass Manufacturing of Wood-Based Furniture*	*Lasa - Lasa is a one step assembly system. Magnetic-assisted twist-lock fasteners generate a positive user assembly experience that is fast, intuitive, and tactile.*	Fall	CFP/CEP	Critical Experience Prototype	Elegance of Sound	Stanford	-	point-based	yes	X	X				
			Fall	CFP/CEP	Critical Experience Prototype	Elegance of Aesthetic	Stanford	-	point-based	yes	X	X				
			Fall	CFP/CEP	Critical Experience Prototype	Elegance of utility and modularity	Stanford	-	point-based	yes	X	X				
			Fall	CFP/CEP	Critical Function Prototype	New Structure for LACK Table	Partner	monofunctional	point-based	no				X		
			Winter	Dark Horse	Dark Horse Prototype	Klik 1	Stanford	monofunctional	point-based	yes		X		X	X	
			Winter	Dark Horse	Dark Horse Prototype	Klik 3	Stanford	monofunctional	point-based	yes			X	X		
			Winter	Dark Horse	Dark Horse Prototype	Klik Lock	Stanford	monofunctional	point-based	yes				X		
			Winter	Funky	Critical Function Prototype	Klik Twist	Stanford	monofunctional	point-based	no				X		
			Winter	Funky	Critical Function Prototype	Kröss	Stanford	monofunctional	point-based	no				X		
			Winter	Funky	Funky Prototype	Kröss Table	Stanford	multifunctional	point-based	yes	X			X		
			Winter	Functional	Critical Function Prototype	Blackklover	Stanford	monofunctional	point-based	no			X			
			Winter	Functional	Critical Function Prototype	Yalloklover	Stanford	monofunctional	point-based	no			X			
			Winter	Functional	Functional Prototype	Weittklover Table	Stanford	multifunctional	point-based	yes	X		X			
			Winter	Functional	Critical Function Prototype	Hafmün Magnets	Stanford	monofunctional	point-based	no				X		
			Spring	Final	Critical Function Prototype	Hafmün Snaps	Stanford	monofunctional	point-based	no			X	X		
			Spring	Final	Critical Function Prototype	Lasa 3D Printed	Stanford	monofunctional	point-based	no				X		
			Spring	Final	Final Prototype	Lasa	Stanford	multifunctional	point-based	yes	X		X	X		X
2016 Renault	*Enhance the User Experience of Future User of highly autonomous vehicles*	**ReForm** *- ReForm is a table that allows users to work in autonomous cars. The table can quickly be deployed when the car is in autonomous mode or folded away when the user needs to drive. The upper surface of the table is an array of deformable spring-loaded elements that create a programmable surface or mesh, which can locate and cradle common items such as cellphones, books, tablets and writing utensils, preventing them from sliding around during normal driving.*	Fall	CFP/CEP	Critical Function Prototype	See Through display	Stanford	monofunctional	point-based	no				X		
			Fall	CFP/CEP	Critical Experience Prototype	Car Coup container Space	Partner	-	point-based	no		X				
			Fall	CFP/CEP	Critical Experience Prototype	Moveable vs. Fixed Chair	Stanford	-	point-based	yes		X				
			Fall	CFP/CEP	Critical Experience Prototype	Motion Sickness White Bar	Partner	-	point-based	yes		X				
			Winter	Dark Horse	Dark Horse Prototype	Active Balancing Vehicle Dynamics	Stanford	monofunctional	point-based	yes	X					
			Winter	Dark Horse	Dark Horse Prototype	Sleeping in Car	Partner	-	point-based	yes	X	X				
			Winter	Dark Horse	Dark Horse Prototype	Vacuum Blanket A1	Partner	monofunctional	point-based	no	X					
			Winter	Dark Horse	Dark Horse Prototype	Vacuum Blanket A2	Partner	monofunctional	point-based	no	X		X	X		
			Winter	Dark Horse	Dark Horse Prototype	Vacuum Blanket B1	Partner	multifunctional	point-based	yes		X	X	X		
			Winter	Funky	Funky Prototype	Air Cooling Seat 1	Partner	multifunctional	point-based	yes		X		X		
			Winter	Funky	Funky Prototype	Air Cooling Seat 2	Partner	multifunctional	point-based	yes		X		X		
			Winter	Funky	Funky Prototype	Particle Jamming Grip Table	Stanford	multifunctional	point-based	yes	X			X	X	
			Winter	Functional	Functional Prototype	The Mesh	Stanford	multifunctional	point-based	yes	X	X		X		
			Winter	Functional	Functional Prototype	Adjustable Table	Stanford	multifunctional	point-based	yes	X					
			Spring	Final	Critical Experience Prototype	Positions PT	Partner	-	point-based	yes		X				
			Spring	Final	Critical Function Prototype	Snap-Fit PT	Stanford	monofunctional	parallel	no				X		
			Spring	Final	Critical Function Prototype	Magnet PT	Stanford	monofunctional	parallel	no				X		
			Spring	Final	Final Prototype	ReForm	Stanford	multifunctional	point-based	yes	X			X		X

Project (free description)	Project prompt (free description)	Final product (free description)	Quarter (Fall; Winter; Spring)	Development phase (CFP/CEP; Dark Horse; Funky; Functional; Final)	Type of Prototype (Critical Function PT; Critical Experience PT; Dark Horse PT; Funky PT; Functional PT; Final PT)	Name of Prototype (free description)	Built by (Stanford / Partner university)	Functionality (monofunctional / multifunctional)	Development mode (point-based/ parallel)	Tested with external users? (yes / no)	Communicate	Explore	Refine	Technically Verify	problem formulation fixed	concept fixed
2016 ShoeInn	*Redesigning the Transition between Clean environments*	*ShoeInn Fit - ShoeInn Fit is a Shoe Cover Dispenser which fits custom sized shoe covers over the person's shoes, as they enter a cleanroom facility.*	Fall	CFP/CEP	Critical Function Prototype	Shoe Cover Remover - Scratch	Stanford	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Shoe Cover Remover - Pinch Method V1	Stanford	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Shoe Cover Remover - Pinch Method V2	Stanford	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Shoe Cover Remover - Claw	Stanford	monofunctional	parallel	no			x	x		
			Fall	CFP/CEP	Critical Function Prototype	Shoe Cover Remover - Roller	Stanford	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Experience Prototype	Clean Room CEP	Stanford	-	point-based	yes		x				
			Fall	CFP/CEP	Critical Experience Prototype	Project Runway	Partner	-	point-based	yes		x				
			Fall	CFP/CEP	Critical Function Prototype	Stardust Yellow	Partner	monofunctional	point-based	no				x		
			Fall	CFP/CEP	Critical Experience Prototype	Space instructor	Partner		point-based	yes	x					
			Winter	CFP/CEP	Critical Experience Prototype	Poncho Transformation Body Size	Stanford	-	point-based	yes		x				
			Winter	CFP/CEP	Critical Experience Prototype	Robot Bunny Suit	Stanford	-	point-based	yes		x				
			Winter	Dark Horse	Dark Horse Prototype	The Dress	Partner	monofunctional	point-based	yes		x				
			Winter	Dark Horse	Dark Horse Prototype	Amazing Elastic Plastic	Stanford	monofunctional	point-based	yes	x	x			x	
			Winter	Dark Horse	Dark Horse Prototype	The 90's Bell bottom Jeans	Partner	monofunctional	point-based	yes		x				
			Winter	Dark Horse	Dark Horse Prototype	The Pyramid	Partner	monofunctional	point-based	yes		x				
			Winter	Dark Horse	Dark Horse Prototype	Spray on Shoes	Stanford	monofunctional	point-based	yes	x					
			Winter	Dark Horse	Dark Horse Prototype	Full Body Suit	Partner	monofunctional	point-based	yes		x				
			Winter	Funky	Funky Prototype	Sealing Gates Prototype V1	Stanford	monofunctional	point-based	no				x		
			Winter	Funky	Funky Prototype	Sealing Gates Prototype V2	Stanford	multifunctional	point-based	no			x	x		
			Winter	Funky	Funky Prototype	Opening Shoe cover - Side Tape	Partner	multifunctional	point-based	no				x		
			Winter	Funky	Funky Prototype	Opening Shoe Cover - Flexible Belt	Partner	multifunctional	point-based	no				x		
			Winter	Functional	Functional Prototype	Sealing Gates System Experience PT	Stanford	multifunctional	point-based	yes	x	x	x			
			Winter	Functional	Functional Prototype	Sealing Gates System Platform PT	Stanford	multifunctional	point-based	yes			x			
			Winter	Functional	Functional Prototype	Final Functional Sealing Gates	Stanford	multifunctional	point-based	yes			x	x		x
			Winter	Functional	Functional Prototype	Ferris Wheels	Partner	multifunctional	point-based	yes			x			
			Winter	Functional	Functional Prototype	Train Rides	Partner	multifunctional	point-based	no			x			
			Winter	Functional	Functional Prototype	Bungee Jump	Partner	multifunctional	point-based	yes			x			
			Spring	Final	Critical Function Prototype	Guiding Prototype	Stanford	monofunctional	point-based	yes		x		x		
			Spring	Final	Critical Function Prototype	Part X Guiding	Stanford	monofunctional	point-based	no			x			
			Spring	Final	Final Prototype	Shoe Inn Fit	Stanford	multifunctional	point-based	yes	x		x			
2016 Ford	Find a method of alternative transportation that would be acceptable to pedestrian motion and track, possibly leveraging autonomous vehicle technology	*CarGo - Cargo is a motorized hand truck with adjustable handles that reduces the effort of the user when transporting heavy loads. For assisting the user during retrieval and storage of the hand truck, a storage system that integrates with the rear door of a Ford Transit was developed.*	Fall	CFP/CEP	Critical Experience Prototype	Tactile 1	Partner	-	parallel	yes	x	x				
			Fall	CFP/CEP	Critical Experience Prototype	Tactile 2	Partner	-	parallel	yes	x	x				
			Fall	CfP/CEP	Critical Experience Prototype	Tactile 3	Partner	-	parallel	yes	x	x				
			Fall	CfP/CEP	Critical Function Prototype	Machine Foldability	Stanford	monofunctional	point-based	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Nestability Foldability	Stanford	monofunctional	point-based	no				x		
			Winter	CFP/CEP	Critical Experience Prototype	Simulate Autonomous Vehicle	Partner		point-based	yes		x				
			Winter	CfP/CEP	Critical Experience Prototype	Information Present	Stanford	-	point-based	yes	x	x				
			Winter	Dark Horse	Dark Horse Prototype	Taco	Stanford	monofunctional	point-based	yes				x	x	
			Winter	Dark Horse	Dark Horse Prototype	Dolly with Brakes	Partner	monofunctional	point-based	no				x		
			Winter	Funky	Funky Prototype	PIE	Stanford	multifunctional	point-based	yes	x	x		x		
			Winter	Funky	Funky Prototype	Motorized Rear Wheels Dolly	Partner	multifunctional	point-based	no				x		
			Winter	Functional	Functional Prototype	Milk	Stanford	multifunctional	point-based	yes		x		x		
			Winter	Functional	Functional Prototype	Ergonomic Problem	Partner	monofunctional	point-based	yes				x		
			Spring	Final	Critical Function Prototype	Haptic Handle Components	Stanford	monofunctional	point-based	no				x		
			Spring	Final	Critical Function Prototype	Haptic Handlebar Force Sensetive	Stanford	monofunctional	point-based	no			x	x		
			Spring	Final	Final Prototype	CarGo	Stanford	multifunctional	point-based	yes	x					x

Project (free description)	Project prompt (free description)	Final product (free description)	Quarter (Fall; Winter; Spring)	Development phase (CFP/CEP; Dark Horse; Funky; Functional; Final)	Type of Prototype (Critical Function PT; Critical Experience PT; Dark Horse PT; Funky PT; Functional PT; Final PT)	Name of Prototype (free description)	Built by (Stanford / Partner university)	Functionality (monofunctional / multifunctional)	Development mode (point-based/ parallel)	tested with external users? (yes / no)	Communicate	Explore	Refine	Technically Verify	problem formulation fixed	concept fixed
2014 Mabe	*Innovate the infridge ice maker*	***Polar Roller*** *- The Polar Roller is a device that can cool down beverage cans from room temperature to an ideal drinking temperature in less than a minute. Furthermore, the device is able to freeze one liter of water in about ten minutes.*	Fall	CFP/CEP	Critical Function Prototype	Rankine Vortex Simulation	Stanford	monofunctional	parallel	no				x	x	
			Fall	CFP/CEP	Critical Function Prototype	Mill Chill	Stanford	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Copper Coil External Cooling	Stanford	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Compressed Gas Cooling	Stanford	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Liquid Cooling	Partner	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Dry Ice Bucket	Partner	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Vacuum Freezing	Partner	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Vibrating Ice	Stanford	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Water Peruity	Stanford	monofunctional	parallel	no				x		
			Fall	CFP/CEP	Critical Experience Prototype	Social Gathering	Stanford	-	point-based	yes		x				
			Fall	CFP/CEP	Critical Experience Prototype	Ice Lucho	Stanford	-	point-based	yes		x				
			Winter	Dark Horse	Dark Horse Prototype	Freeze Testing	Stanford	monofunctional	point-based	no	x			x		
			Winter	Dark Horse	Dark Horse Prototype	Mr. Freeze	Stanford	monofunctional	point-based	yes	x					
			Winter	Dark Horse	Dark Horse Prototype	Pokernight	Stanford	monofunctional	point-based	yes	x					
			Winter	Funky	Funky Prototype	Polar Roller 1.0	Stanford	multifunctional	point-based	no				x		x
			Winter	Functional	Functional Prototype	Polar Roller 2.0	Stanford	multifunctional	point-based	no			x	x		
			Winter	Functional	Critical Function Prototype	Ice Skeleton 1.0	Stanford	monofunctional	point-based	no				x		
			Winter	Functional	Critical Function Prototype	Ice Skeleton 2.0	Stanford	monofunctional	point-based	no			x			
			Spring	Final	Final Prototype	Polar Roller 3.0	Stanford	multifunctional	point-based	yes	x		x			
2014 VolvoCE	*Develop Future Equipment for Urban Mining*	***Volvo 310x*** *- The Volvo 310X is a prototype concrete planing machine that is designed to more effectively remove concrete and facilitate the safe and efficient transport of the concrete debris from the work area. The machine is remotely controlled and removes concrete flooring layer by layer.*	Fall	CFP/CEP	Critical Function Prototype	Rapid Prototype 1	Stanford	monofunctional	parallel	no	x					
			Fall	CFP/CEP	Critical Function Prototype	Rapid Prototype 2	Stanford	monofunctional	parallel	no	x					
			Fall	CFP/CEP	Critical Function Prototype	Rapid Prototype 3	Stanford	monofunctional	parallel	no	x					
			Fall	CFP/CEP	Critical Function Prototype	Pipe Cutter Prototype	Stanford	monofunctional	point-based	no				x		
			Fall	CFP/CEP	Critical Function Prototype	Final Pipe Cutter Prototype	Stanford	monofunctional	point-based	no			x			
			Winter	Dark Horse	Dark Horse Prototype	iRobot with Bucket	Stanford	multifunctional	point-based	yes	x	x				
			Winter	Funky	Funky Prototype	Lego Skid Steer	Stanford	multifunctional	point-based	yes	x	x			x	
			Winter	Functional	Critical Function Prototype	Concrete Breaking Jack Hammers	Stanford	monofunctional	point-based	no				x		
			Winter	Functional	Functional Prototype	Concrete Chipper Cart 1	Stanford	multifunctional	point-based	no				x		
			Winter	Functional	Functional Prototype	Concrete Chipper Cart 2	Stanford	multifunctional	point-based	yes	x		x			
			Spring	Final	Final Prototype	Volvo 310x	Stanford	multifunctional	point-based	yes	x		x			x

Appendix C: analysis of the connection between prototypes and the development of problem and solution space

Type of prototype connection	Description	Graphic representation
Concept connection	Previous prototype and subsequent prototype share the same concept (e. g. Further development / improvement of a prototype).	
Trigger connection	**Adjustment in solution space:** Previous prototype triggers a concept change of the subsequent prototype. Both prototypes have the same underlying problem formulation.	
	Adjustment in problem space: Previous triggers a change in problem formulation of the subsequent prototype (automatically entails a concept change of subsequent prototype).	
Hopping	Subsequent prototype tackles a different problem space without any discernible influence of the previous prototype.	

Project	Quarter (Fall; Winter; Spring)	Development phase (CFP/CEP; Dark Horse; Funky; Functional; Final)	PT X - PT Y (CFP; CEP; DH; FKP; FSP; FP)	PT X - PT Y (CFP; CEP; DH; FKP; FSP; FP)	Name PT X	Name PT Y	Logic (Concept connection; Trigger connection; Hopping)	Change in problem formulation	Concept change	Change triggered (internally; externally; not specified)	Reason for change in problem or solution space
2019 Volkswagen	Fall	CFP/CEP	CFP	CFP	Particle Jamming	Particle Jamming Scale Up	Concept connection			-	-
	Winter	Dark Horse	CEP	DH	SEDRIC Mockup	Scooper Door	Trigger connection	X	X	externally	For our Team, a prototype focuse on something other than the securement of the wheelchair would be a dark prototype because it is tackling a different problem space than the one we had been focused on all of Fall quarter
	Winter	Dark Horse	CEP	DH	SEDRIC Mockup	Collapsible Storage for Service Animals	Trigger connection	X	X	externally	CFP feedback
	Winter	Dark Horse	CEP	DH	SEDRIC Mockup	Moka	Trigger connection	X	X	externally	Explore alternative entering concept
	Winter	Funky	CFP	FKP	Particle Jamming Scale UP	Particle Jammed Clamps	Concept connection			-	-
	Winter	Funky	CFP	FKP	Fi Particles - Coffee Ground	Particle Jammed Clamps	Concept connection			-	-
	Winter	Functional	FKP	FSP	Particle Jammed Clamps	Automated Voice Activated Clamps	Trigger connection		X	internally	Particle Jamming not worth the added complexity
	Winter	Functional	CEP	FSP	SEDRIC Mockup	Turntable	Trigger connection	X	X	externally	Explore Different Problemspace
	Spring	Final	FSP	CFP	Automated Voice Activated Clamps	Rail System	Trigger connection	X	X	externally	User raised some concerns about entering the vehicle
	Spring	Final	FSP	FP	Automated Voice Activated Clamps	Rise	Trigger connection	X	X	internally	Subsystem integrated
	Spring	Final	CFP	FP	Rail System	Rise	Trigger connection		X	internally	Decided to pivot to a system that uses a robot pod as it could work without any modifications to our accesible van
2019 Panasonic	Fall	CFP/CEP	CFP	CFP	PolyPlug	Poly Plug V2	Concept connection			-	-
	Winter	Dark Horse	-	DH	-	Sensing Mechanism - Pressure Sensing Sensing Mechanism - Vision Sensing	Hopping	X	X	internally	After a more critical analys of Multi Casa, the Design Team realized potential flaws in their vision
	Winter	Dark Horse	-	DH	-	Turtle Home	Hopping	X	X	internally	Dark Horse Hopping
	Winter	Funky	-	CFP	-	Proportion Consideration	Hopping	X	X	not specified	No reason specified
	Winter	Funky	CEP	FKP	Proportion Consideration	Tot Wrangler	Concept connection			-	-
	Winter	Functional	FKP	FSP	Soft Robot	Scissor Gate	Trigger connection		X	internally	Soft Robot was to complex and Scissor Gate is a simpler option
	Spring	Final	-	CFP	-	E-Scooter Connector It.1	Hopping	X	X	not specified	No reason specified -> Development E-Scooter Connector
	Spring	Final	CFP	CFP	E-Scooter Connector It.1	E-Scooter Connector Pin	Trigger connection		X	internally	
	Spring	Final	CFP	CFP	E-Scooter Connector Pin	Snap-Fit Connection PT	Trigger connection		X	internally	Pin Design flawed in a few ways. Difficult to ensure the pin. Better Idea to connect to the metal frame covering the caster than the wheel
	Spring	Final	CFP	CFP	Snap-Fit Connection PT	Pen Ult Connector	Trigger connection		X	internally	New design because they cant develop the 3 wheeled stroller envisiond
	Spring	Final	CFP	FP	Pen Ult Connector	Koko Mover	Concept connection			-	-
2018 Safran	Fall	CFP/CEP	CEP	CFP	The Useful Wheel Chair	Generator Wheelchair	Concept connection			-	-
	Winter	Dark Horse	-	DH	-	Rack and Pinion Rail System	Hopping	X	X	internally	Change Problemspace to public transportation
	Winter	Funky	-	FKP	-	Wheelchair Bike Rack	Hopping	X	X	internally	Ascending and descending require some infrastructure changes to the sairs, namely a secure attachment point Due to these problems we shifted our focus away from public transportation towards improving personal transportation with vehicles
	Winter	Funky	FKP	FKP	Wheelchair Bike Rack	Wheelchair Bike Rack refined	Concept connection			-	-
	Winter	Funky	FKP	FKP	Wheelchair Bike Rack refined	Wheelchair Bike Rack + Cover	Concept connection			-	-
	Winter	Functional	FKP	FSP	Wheelchair Bike Rack + Cover	Wheelchair Transit Adapter	Concept connection			-	-
	Winter	Final	FSP	FP	Wheelchair Transit Adapter	Ryde & Slyde	Trigger connection		X	internally	moved system to the back of the trunk
2017 Audi	Fall	CFP/CEP	CEP	CFP	Car Handover	Car Handover Speech In and Output	Concept connection			-	-
	Winter	Funky	DH	FKP	Hitchhiking Robot	Smart Beacon	Trigger connection		X	internally	-
	Winter	Functional	-	FSP	-	Carrus System	Hopping	X	X	internally	No Use Case for FKP found
	Spring	Final	-	CFP	-	Magnet Mount Suction Mount	Hopping	X	X	internally	User Observation in Germany
	Spring	Final	CFP	CFP	Magnet Mount	Magnet Cable Config	Trigger connection		X	internally	Magnets couldnt hold due to shear forces -> Switched to magnets + flexible cable
	Spring	Final	CFP	FP	Magnet Cable Config	Hitch	Concept connection			-	-
2016 IKEA	Winter	Dark Horse	-	DH	-	Klik 1	Hopping	X	X	internally	Explore Assembly Mechanism after Fall Quarter
	Winter	Dark Horse	DH	DH	Klik 1	Klik 3	Concept connection			-	-
	Winter	Dark Horse	DH	DH	Klik 3	Klik Lock	Trigger connection		X	internally	unsatisfied with the rigidity issues faced by klik 3, we replaced the fabric snaps with magnets
	Winter	Funky	CFP	CFP	Klik Twist	Krõss	Trigger connection		X	internally	expand on learning from Klik Twist 1, most notably that snaps and magnets were not necessary, that a more robust alignment mechanism was necessary, and that we wanted to integrate our mechanism into a fuctional furniture product
	Winter	Funky	CFP	FKP	Krõss	Krõss Table	Concept connection			-	-
	Winter	Functional	FKP	CFP	Krõss Table	Blackklover	Concept connection			-	-
	Winter	Functional	CFP	CFP	Blackklover	Yelloklover	Concept connection			-	-
	Winter	Functional	CFP	FKP	Yelloklover	Weittklover Table	Concept connection			-	-
	Spring	Final	CFP	CFP	Hafmün Magnets	Hafmün Snaps	Trigger connection		X	externally	Dicussion with IKEA Industry and IKEa corporation
	Spring	Final	CFP	CFP	Lasa 3D printed	Lasa	Concept connection			-	-
2016 Renault	Fall	CFP/CEP	-	CEP	-	Moveable vs. Fixed Chair	Hopping	X	X	internally	-
	Fall	CFP/CEP	-	CEP	-	Motion Sickness White Bar	Hopping	X	X	internally	-
	Winter	Dark Horse	CEP	DH	Moveable vs. Fixed Chair	Active Balancing Vehicle Dynamics	Trigger connection		X	internally	Explore the Dark Side
	Winter	Dark Horse	-	DH	-	Sleeping in Car	Hopping	X	X	internally	Motion sickness is not a approriate Topic to Tackle in ME310
	Winter	Dark Horse	DH	DH	Vacuum Blanket A1	Vacuum Blanket A2	Concept connection			-	-
	Winter	Dark Horse	DH	DH	Vacuum Blanket A2	Vacuum Blanket B1	Concept connection			-	-
	Winter	Funky	-	FKP	-	Air Cooling Seat 1	Hopping	X	X	externally	Coporate Sponsor Advised against continuing with
	Winter	Funky	DH	FKP	Active Balancing Vehicle Dynamics	Particle Jamming Grip Table	Trigger connection	X	X	externally	Things Fall from the Table
	Winter	Funky	FKP	FKP	Air Cooling Seat 1	Air Cooling Seat 2	Concept connection			-	-
	Winter	Functional	FKP	FSP	Particle Jamming Grib Table	The Mesh	Trigger connection		X	internally	Particle Jamming dont work well -> inspired by MIT
	Winter	Functional	-	FSP	-	Adjustable Table	Hopping	X	X	internally	Converged on one Design Vision with Stanford Team
	Spring	Final	CFP	FP	Snap-Fit PT	ReForm	Trigger connection		X	internally	Combined Both CFPs
	Spring	Final	CFP	FP	Magnet PT	ReForm	Trigger connection		X	internally	Combined Both CFPs
	Spring	Final	CEP	FP	Positions PT	ReForm	Concept connection			-	
	Spring	Final	FSP	FP	The Mesh	ReForm	Trigger connection		X	-	-

Project	Quarter (Fall; Winter; Spring)	Development phase (CFP/CEP; Dark Horse; Funky; Functional; Final)	PT X - PT Y (CFP; CEP; DH; FKP; FSP; FP)	PT X - PT Y (CFP; CEP; DH; FKP; FSP; FP)	Name PT X	Name PT Y	Logic (Concept connection; Trigger connection; Hopping)	Change in problem formulation	Concept change	Change triggered (internally; externally; not specified)	Reason for change in problem or solution space
2016 Shoelos	Fall	CFP/CEP	CFP	CFP	Shoe Cover Remover - Pinch Method V1	Shoe Cover Remover - Pinch Method V2	Concept connection			-	-
	Fall	Dark Horse	-	DH	-	Amazing Elastic Plastic	Hopping	X	X	internally	-
	Winter	Funky	-	FKP	-	Opening Shoe cover - Side Tape	Hopping	X	X	internally	not specified
	Winter	Funky	DH	FKP	Amazing Elastic Plastic	Sealing Gates Prototype V1	Trigger connection		X	internally	Idea is the same but make it funky
	Winter	Funky	FKP	FKP	Sealing Gates Prototype V1	Sealing Gates Prototype V2	Concept connection			-	-
	Winter	Funky	FKP	FKP	Opening Shoe Cover - Side Tape	Opening Shoe Cover - Flexible Belt	Trigger connection		X	internally	explore new mechanical solutions
	Winter	Functional	FKP	FSP	Opening Shoe Cover - Flexible Belt	Ferris Wheels	Concept connection			-	-
	Winter	Functional	FKP	FSP	Sealing Gates Prototype V2	Sealing Gates System Experience PT	Concept connection			-	-
	Winter	Functional	FSP	FSP	Sealing Gates System Experience PT	Sealing Gates System Platform PT	Concept connection			-	-
	Winter	Functional	FSP	FSP	Sealing Gates System Platform PT	Final Functional Sealing Gates	Trigger connection		X	internally	Different Loading System, different Guiding System,
	Winter	Functional	FSP	FSP	Ferris Wheels	Train Rides	Concept connection			-	-
	Winter	Functional	FSP	FSP	Train Rides	Bungee Jump	Concept connection			-	-
	Spring	Final	CFP	CFP	Guiding Prototype	Part X Guiding	Concept connection			-	-
	Spring	Final	CFP	FP	Part X Guiding	Shoe Inn Fit	Concept connection			-	-
	Spring	Final	FSP	FP	Final Functional Sealing Gates	ShoeInn Fit	Concept connection			-	-
2016 Ford	Winter	Dark Horse	-	DH	-	Taco	Hopping	X	X	internally	-
	Winter	Dark Horse	-	DH	-	Dolly with Brakes	Hopping	X	X	internally	-
	Winter	Funky	DH	FKP	Taco	PIE	Trigger connection		X	internally	Improve TACO - Self Stabalizing Dolly
	Winter	Functional	FKP	FSP	PIE	Milk	Concept connection			-	-
	Winter	Functional	FKP	FSP	Motorized Rear Dolly	Milk	Concept connection			-	-
	Spring	Final	FSP	FP	Milk	CarGo	Trigger connection		X	internally	
	Spring	Final	CFP	CFP	Haptic Handle Components	Haptic Handlebar Force Sensetive	Trigger connection		X	internally	
	Spring	Final	CFP	FP	Haptic Handlebar Force Sensetive	CarGo	Concept connection			-	-
2014 Mabe	Winter	Functional	CFP	CFP	Ice - Skeleton 1.0	Ice Skeleton 2.0	Concept connection			-	-
	Winter	Functional	FKP	FSP	Polar Roller 1.0	Polar Roller 2.0	Concept connection			-	-
	Spring	Final	FSP	FP	Polar Roller 2.0	Polar Roller 3.0	Concept connection			-	-
	Spring	Final	CFP	FP	Ice - Skeleton 2.0	Polar Roller 3.0	Concept connection			-	-
2014 Volvo CE	Fall	CFP/CEP	CFP	CFP	Pipe Cutter Prototype	Pipe Cutter Prototype Final	Concept connection			-	-
	Winter	Dark Horse	-	DH	-	iRobot with Bucket	Hopping	X	X	internally	Change in problem space
	Winter	Funky		FKP	-	Lego Skid Steer	Hopping	X	X	externally	Change in problem space
	Winter	Functional	CFP	FSP	Concrete Breaking Jack Hammers	Concrete Chipper Cart 1	Concept connection			-	-
	Winter	Functional	FSP	FSP	Concrete Chipper Cart 1	Concrete Chipper Cart 2	Concept connection			-	-
	Spring	Final	FSP	FP	Concrete Chipper Cart 2	Volvo 310s	Trigger connection		X	internally	Wheels -> Chain, different system for removing concrete