

THE CHALLENGING COMBINATION OF AGILITY AND CONVERGENCE IN HYBRID PRODUCT DEVELOPMENT PROCESSES: AN EMPIRICAL ANALYSIS OF STANFORD'S ME310 PROCESS MODEL

Kopenhagen, Frank (1);
Blümel, Tim (2);
Held, Tobias (1);
Wecht, Christoph (3);
Kollmer, Paul Davin (4)

1: Hamburg University of Applied Sciences;
2: University of Stuttgart;
3: New Design University St. Pölten;
4: University of Hamburg

ABSTRACT

Combining agility and convergence in the development of physical products is a major challenge. Rooted in a design thinking approach, Stanford's ME310 process model attempts to resolve the conflicting priorities of these two design principles. To investigate how successful Stanford's hybrid process model is in doing so, we have used a qualitative case study approach. Our paper begins by outlining this process model's fundamental principles in terms of engineering design methodology. Subsequently, we present the results of our empirical analysis, which tracks the coevolution of problem and solution space by meticulously examining all prototype paths in ten of Stanford's ME310 student projects. We have discovered that convergence during solution finding does not correspond to the process model's theoretical specifications. Even in the phase of the final prototype, both the technical concept and the underlying problem formulation changed frequently. Further research should focus on combining the prototype-based ME310 approach with methods from systems engineering which allow for a more comprehensive theoretical exploration of the solution space. This could lead to improved convergence during solution development.

Keywords: Design engineering, Design methodology, Case study, Design thinking, Agile product development

Contact:

Kopenhagen, Frank
Hochschule für Angewandte Wissenschaften Hamburg
Maschinenbau und Produktion
Germany
frank.kopenhagen@haw-hamburg.de

Cite this article: Kopenhagen, F., Blümel, T., Held, T., Wecht, C., Kollmer, P. D. (2021) 'The Challenging Combination of Agility and Convergence in Hybrid Product Development Processes: An Empirical Analysis of Stanford's ME310 Process Model', in *Proceedings of the International Conference on Engineering Design (ICED21)*, Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.560

1 INTRODUCTION

Although previous studies have demonstrated that it is a suitable approach for the user-centered development of physical products (Brown, 2008; Luchs *et al.*, 2016), design thinking has thus far received little attention in the academic discourse of engineering design methodology. The limited number of studies analyzing 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. The inapt approach of previous studies can partly be attributed to the lack of a consistent definition of design thinking, which makes it difficult to classify the concept with regard to engineering design methodology. But it is also due to the fact that most design thinking process models do not encompass an entire product development process. An exception is Stanford's ME310 process model. Based on an agile design thinking approach, this hybrid model also defines milestones to guide the convergence of solution development.

The purpose of this paper is to investigate the practical suitability of the ME310 process model for the development of physical products. Since it aims to combine agility and convergence in the realm of New Product Development, 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 analyzed ten 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. The results of our explorative study reveal that the development paths observed in practice do not follow the convergence path the process model specifies. Deviating considerably from the ME310 process model, both underlying problem formulations and technical concepts changed frequently in the late stages of the product development process.

2 THEORETICAL FRAMEWORK

2.1 Problem and solution space in product development

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, focused 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.

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 operationalized by formulating requirements before the actual solution can be developed, which, in turn, is finally intended to satisfy the actual user's need. 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 coevolution of problem and solution space (Dorst and Cross, 2001) is guided by user interaction tests in which prototypes embody certain aspects of the evolving product to understand user needs and derive design requirements.

2.2 Semantic levels and methodological taxonomy levels of design thinking

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 1).

The fundamental level describes the action-guiding principles that are constitutive for the mindset and culture of the design thinking approach. 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). 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, synchronize 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.

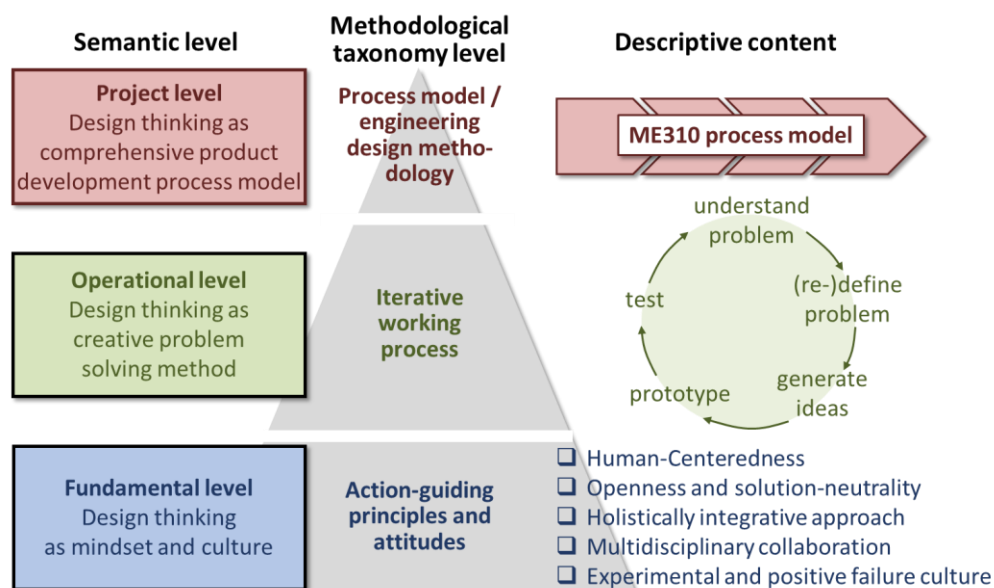


Figure 1: Semantic levels and methodological taxonomy levels of design thinking

The only process model of which we are aware that meets these criteria is the ME310 process model developed at the Center for Design Research at Stanford University. 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.). 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.

2.3 The ME310 process model

In Figure 2, the presentation of each individual phase, along with their assigned activities and intended results, is essentially based on the ME310 ABC Course Reader (Kenyon *et al.*, n.d.). After extensive problem space exploration in the Needfinding phase (NF), resulting in an initial problem formulation, the analysis of existing products during the Benchmarking phase (BM) already establishes a connection between problem and solution space. With this connection, a product vision develops. Then, actual solution development begins in the CEP/CFP phase at the subsystem level. On the one hand, 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, without having already developed a technical function carrier, is simulated. On the other hand, Critical Function Prototypes (CFP) 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.

	Phases	Activities	Results
	Needfinding	User-centered problem space exploration Identifying users, user needs and use context	Initial problem formulation
	Benchmarking	Establishing a connection between problem and solution space Analysis of strengths and weaknesses of existing products addressing the initial problem formulation	Product vision
Subsystem	CEP / CFP	Coevolution of problem and solution space CEP: Derive and validate requirements CFP: Verify suitability of effective principles	Requirements (definitive problem formulation)
	Dark Horse Prototype	Scrutinizing the understanding of the problem Building and testing risky and radically new solutions on the edge or outside of the preliminary solution space	Validated problem formulation
Overall System	Funky System Prototype	Defining and testing the overall concept Combination of CFPs and possibly also DHPs to an overall solution concept, proof of concept	Conceptual design (principle solution)
	Functional System Protot.	Defining and testing the overall design Completion of functional scope, functional optimization, consolidation of system integration, aesthetic design	Embodiment design (geometric layout)
	Final Prototype	Embodiment of the final product Representation of the complete user experience, detail design, refinement, manufacturing plan, final documentation	Complete product representation

Figure 2: Phases, activities and results of the ME310 process model¹

The subsequent phase of the Dark Horse Prototype (DHP) is intended (1) to validate the acquired problem formulation, which, in other words, depicts the understanding of the problem, and (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 Dark Horse prototypes forces the development team to abandon an underlying cognitive solution fixation (Domingo *et al.*, 2020) and scrutinize previous understandings of the problem. This phase should result in a validated problem formulation, generating a firm and reliable framework for subsequent solution development on the overall system level. 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 Funky System Prototype 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 Funky System Prototype. In the subsequent Functional System Prototype (FCP) phase, the concept of the Funky System Prototype is detailed and optimized. The Functional System Prototype is intended to represent the complete functional scope and serves to consolidate system integration as well as optimization on an overall and subfunction level. It defines the embodiment design and should already have a value proposition comparable to the Final Prototype. Lastly, the Final Prototype, marking the completion of development, should represent the complete user experience of a product to be industrially realized.

¹ 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 realized 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 crystallization of the Final Prototype. One can therefore regard the Part-X-is-finished prototype as an intermediate milestone within the Final Prototype phase.

2.4 Fundamental principles of the ME310 process model

In this section, we explain the two fundamental principles of the ME310 process model from the perspective of engineering design methodology. These are: (1) the coevolution of problem and solution space and (2) prototype-based solution development.

2.4.1 Coevolution of problem and solution space

While the systems engineering approach usually clearly separates problem analysis and solution development, the ME310 process model is characterized by a concomitant development of these two spheres in the early prototype stage (Figure 3). For this purpose, two parallel control loops are established in the CEP/CFP phase: a validation and a verification control loop (Figure 4). The user interaction tests based on the CEPs deepen the understanding of the problem and enable the derivation and validation of requirements, some of which only become apparent in practical testing (Leifer and Steinert, 2011). The CEPs thus establish a validation control loop that prevents a “mismatch” between product features and user needs.

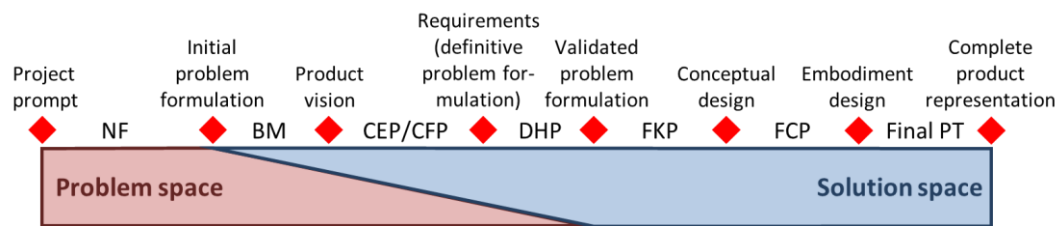


Figure 3: Coevolution of problem and solution space in the ME310 process model

At the same time, the CFPs ensure the suitability of innovative effective principles and the fulfillment 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.

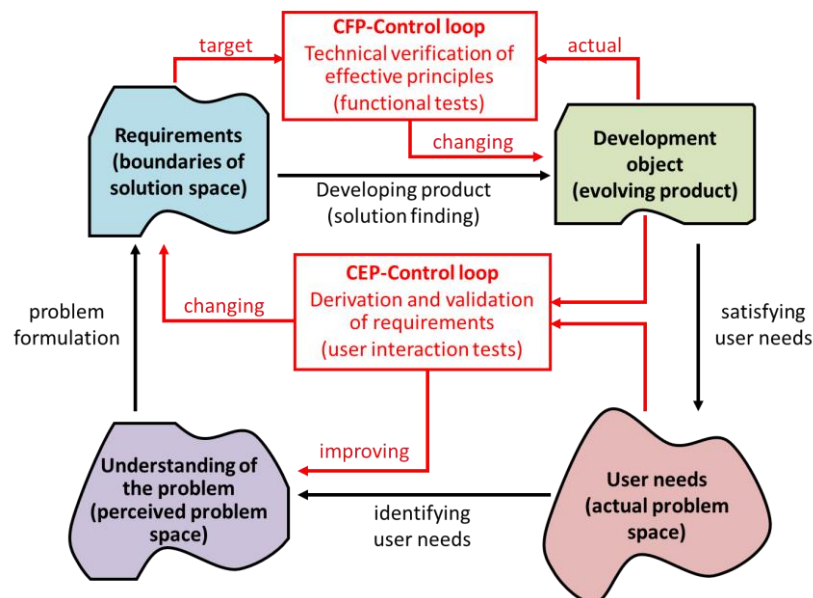


Figure 4: Validation and verification control loop through CEP and CFP

In contrast to systems engineering, the problem formulation in the ME310 process model, meaning the formulation of requirements representing and operationalizing 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 Dark Horse Prototypes. Both approaches nonetheless pursue the same goal: avoiding a premature and overly narrow limitation of the solution space through an inadequate problem formulation.

2.4.2 Prototype-based solution development

Building prototypes is integral to the ME310 design methodology; they are the main drivers of the development process. Within all phases, prototypes are not only used for verification and optimization but also for exploring and communicating solutions. The different kinds of prototypes define the process model's milestones, which structure the overall development process.

Since the realization of a complex physical product involves considerable coordinated effort, development at the overall system level is rarely executed in a completely agile manner. In the ME310 process model, development is therefore only completely agile during the subfunction prototype phases (CEP/CFP and DHP), in which the understanding of both the problem and, consequently, the boundary of the solution space remains volatile. During the system prototype phases, on the other hand, the aim is to achieve increasing convergence and consolidation of the development result, which is controlled by supplementing and detailing requirements, thus leading to an increasingly narrow solution space (Figure 5). From the perspective of design methodology, the ME310 process model is therefore an agile stage-gate hybrid process model (cf. Cooper and Sommer (2016)).

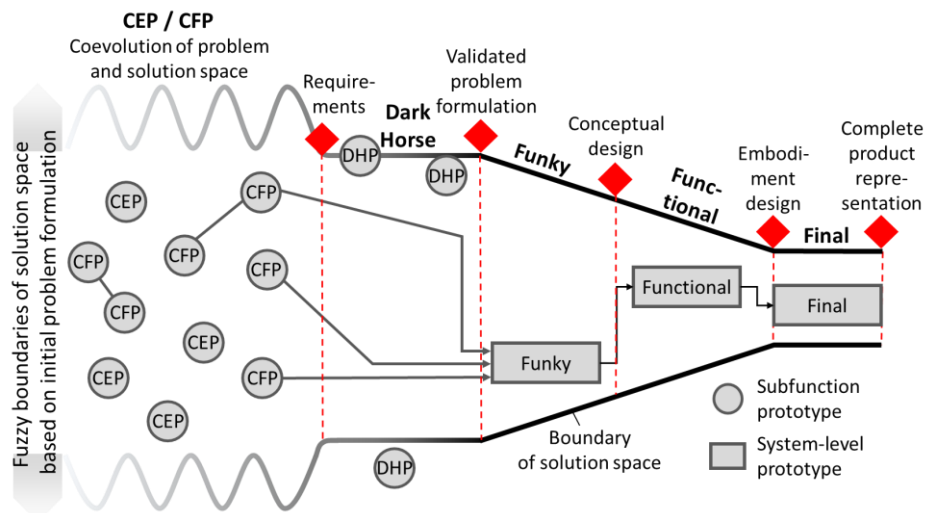


Figure 5: Theoretical convergence path of the ME310 process model

Early subfunction prototypes are usually built within the ME310 process without prior virtual development, whereas intensive virtual development usually starts before subsequent system prototypes are built. For the process of creative solution development, the early use of prototypes brings with it numerous advantages (Brereton and McGarry, 2000; Viswanathan and Linsey, 2012). However, with respect to solution space exploration, a prototype-based approach is a double-edged sword. Development in systems engineering 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" (Sobek II et al., 1999) approach. Yet, within the ME310 development methodology, development is based on prototyping right from the start. Although this generates greater gains in knowledge with regard to the specific embodied solution than a purely virtual development, early prototyping involves considerably more effort, especially on the level of overall system prototypes. Parallel development in the ME310 process model is therefore only intended for the level of the subfunction prototypes. Competing concepts on the overall system level are neither built nor evaluated because parallel system-level prototyping would involve a prohibitive amount of effort. In this respect, the approach of the ME310 design methodology is even

more point-based than the traditional systems engineering design methodology. The solution space explored at the time of concept decision is smaller overall, increasing uncertainty in concept selection. 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 modeling 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; and, (2) the identification of compatible combinations of different CFPs to form the first overall solution, the Funky System Prototype. Both have to be addressed implicitly by the development team, a substantial challenge, especially for complex products. Indeed, the modeling 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 (Kopenhagen, 2004). The lack of such an explicit approach to complexity reduction limits the application possibilities of the ME310 development methodology.

3 METHODOLOGY AND SAMPLE

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 ten. 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 the company-specific influence 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 3,578 pages.

From these project reports, we assessed the technical concept of each prototype and its underlying problem formulation. Problem formulations were determined based on verbal descriptions in project reports. To determine the prototypes' technical concepts, we had to analyze both the function carriers' effective principles and how said function carriers are combined to fulfill the solution-determining main functions of the prototype. In order to do so, we evaluated the development artifacts shown in these reports, such as sketches, diagrams, technical drawings, screenshots of 3D CAD models, and photos of the prototypes built. Based on this evidence, we retraced projects' prototype paths and determined when either changes in problem formulations or concept changes in the solution space occurred. Furthermore, we examined whether findings from the evaluation of previous prototypes influenced these changes, and whether said concept changes were initiated by members of the development team, the teaching team, or industry sponsor representatives. The entire selection process and analysis outlined above was conducted independently by two of our senior researchers specializing in industrial product development. If differences in their assessments arose, their results were consolidated with yet another expert's assistance.

4 FINDINGS AND DISCUSSIONS

Figure 6 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. In the following two sections, we present the overall evaluation of problem and solution space development across all ten projects examined.

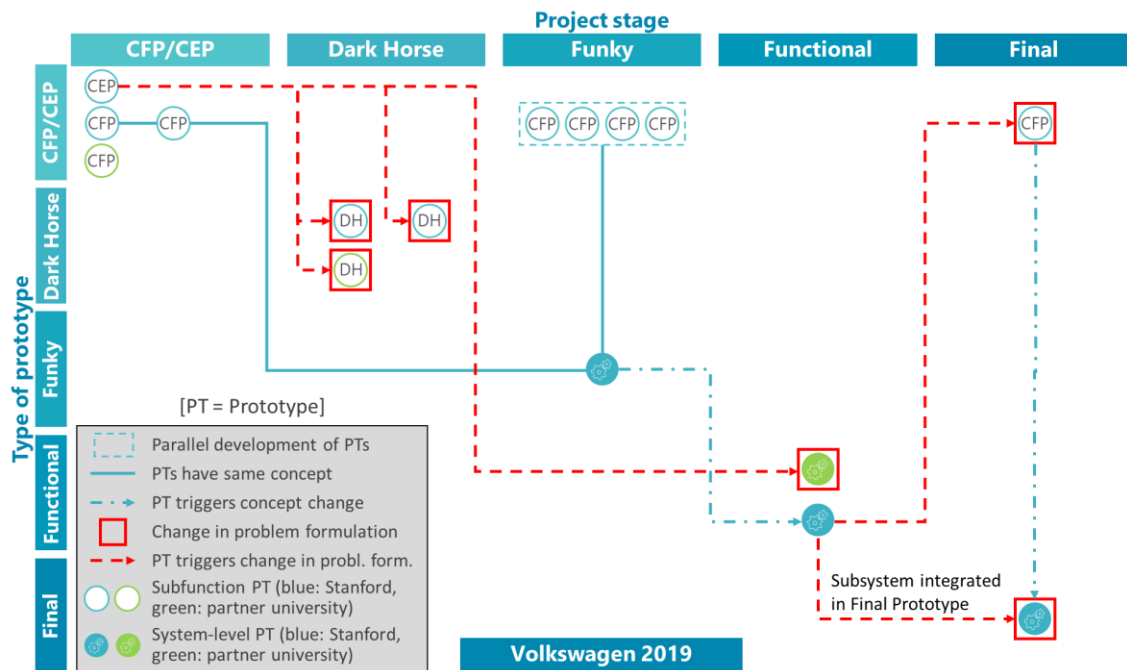


Figure 6: Visualization of the prototype paths of the VW 2019 project

4.1 Development of the problem formulation

Figure 7 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, a phase for which, corresponding to the process model, critical questioning of the previously developed problem formulation is intended. The problem formulation was, however, still frequently modified in the subsequent system prototype phases, in particular in the Funky System Prototype 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 ten 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.

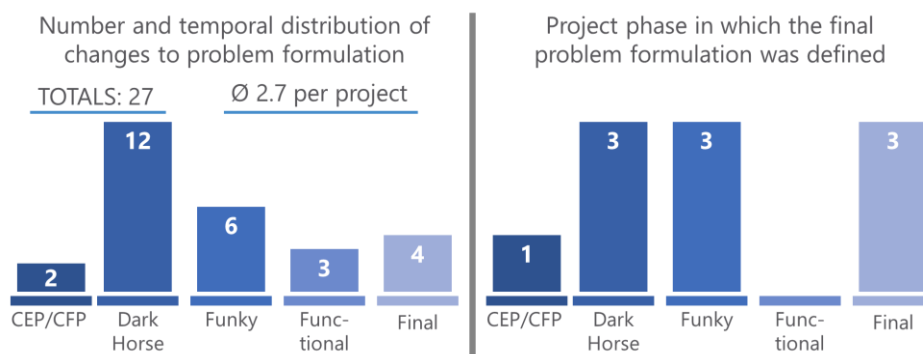


Figure 7: Changes to the problem formulation and the phase of its final definition

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 Funky System Prototype 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.

4.2 Concept development

Figure 8 shows, starting with the Funky System Prototype 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 Funky System Prototype phase is changed frequently during the very same phase. Even after completion of the Funky System Prototype phase, one can observe a high volatility of the technical concepts in play. Remarkably, even in the Final Prototype phase, there were still 16 concept changes spread over eight development projects. 74% 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 Funky System Prototype 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 ten projects was a function carrier, respectively an effective principle, from the CFP phase part of the Funky System Prototype. In all projects investigated, CEPs and/or CFPs are built up again in the later system prototype phases.

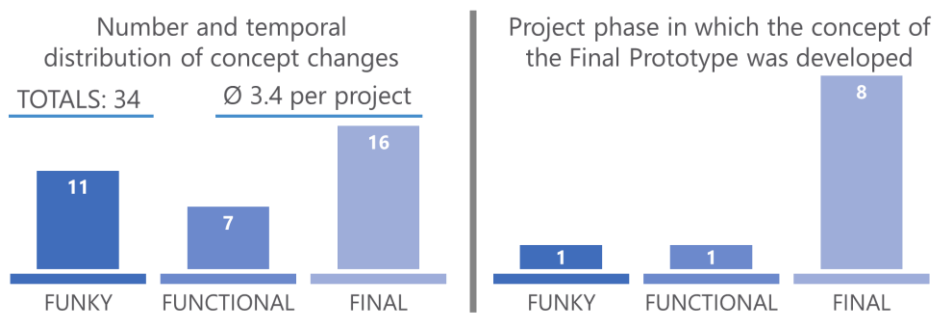


Figure 8: Concept development

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 1.7 times after the final problem formulation was defined.

One reason for the low number of function carriers transferred from the CFP phase to the Funky System Prototype 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 is detrimental to the stability of the concept decision.

5 CONCLUSION AND OUTLOOK

Through in-depth examinations of their project reports, we have tracked the development paths of ten ME310 student projects and compared them with the specifications of the respective process model. The development paths observed do not follow the convergence path which the process model specifies. Changes to both the underlying problem formulations and technical concepts occur frequently in the late stages of the development process. Late convergence during solution development could be attributed to (1) insufficient methodological support of the conceptual design on the overall system level and (2) deficient theoretical exploration of a larger solution space before arrival at a concept decision. Both factors increase the prognostic uncertainty of concept decision, rendering it less stable.

By analyzing additional project reports, further research should validate the findings of our study on a broader empirical basis. In order to address the shortcomings, we have identified, further research might investigate integrating specific methods from the conceptual design phase of systems engineering into the ME310 process model. Based on the theoretical modeling of an overall system with explicit functional decomposition, selected functions could, for example, be represented and tested using CFPs and CEPs. Subsequently, parallel virtual development of multiple concepts could help to explore a larger theoretical solution space. The practical testing of both effective principles and

the user experience associated with certain subfunctions would, at the same time, lead to a more comprehensive understanding of the solution space. This would greatly increase knowledge about the development object before concept selection. A significant challenge, however, is to apply the principle of prototype-based coevolution of the problem and solution space to the overall system level with limited effort, a process only executed at the subsystem level within the ME310 process model. One might face this challenge with the early definition of modular concept architectures upon which different concept variants could be built through the variation and combination of individual solution components. Ultimately, this could lead to a new hybrid process model. To establish the viability of this new process model, a broad empirical analysis is needed which also investigates the influence of different development objects and the organization responsible for the development process's execution.

REFERENCES

- 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*, The Hague, The Netherlands, 01.-06.04.2000, pp. 217–224. <https://doi.org/10.1145/332040.332434>
- Brown, T. (2008), "Design Thinking", *Harvard Business Review*, Vol. 86 No. 6, pp. 84–92.
- 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 & Management for Innovation*, Polo, Portugal, 04.-06.09.2013. <https://doi.org/10.13140/2.1.2361.7602>
- Cooper, R.G. and Sommer, A.F. (2016), "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>
- 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.-29.10.2020, 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](https://doi.org/10.1016/S0142-694X(01)00009-6)
- 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*, Buchholz, Hamburg, Deutschland, 23.-24.09.2010, pp. 117–130.
- Haberfellner, R., Weck, O.L. de, Fricke, E. and Vössner, S. (2019), *Systems Engineering: Fundamentals and Applications*, Springer Nature, Cham, Switzerland. <https://doi.org/10.1007/978-3-030-13431-0>
- Kenyon, D., Cutkosky, M. and Leifer, L. (n.d.), "ME310 ABC Course Reader", Stanford University, Stanford, California, USA.
- Koppenhagen, F. (2004), *Systematische Ableitung modularer Produktarchitekturen: Komplexitätsreduzierung in der Konzeptphase*, Shaker, Aachen.
- Leifer, L.J. and Steinert, M. (2011), "Dancing with ambiguity: Causality behavior, design thinking, and triple-loop-learning", *Information Knowledge Systems Management*, Vol. 10 No. 1-4, pp. 151–173. <https://doi.org/10.3233/IKS-2012-0191>
- Luchs, M.G., Swan, K.S. and Griffin, A. (Eds.) (2016), *Design thinking: New product development essentials from the PDMA*, Wiley, Hoboken, New Jersey.
- Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2007), *Engineering design: A systematic approach*, 3. ed., Springer, London.
- 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.
- 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*, Stuttgart, 16.05.2019, pp. 193–202.
- 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.
- 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>