

# Agile Prototyping for technical systems

## Towards an adaption of the Minimum Viable Product principle

Günther Schuh<sup>1</sup>, Christian Dölle<sup>2</sup>, Sebastian Schloesser<sup>3</sup>

<sup>1,2,3</sup> RWTH Aachen University, Laboratory for Machine Tools and Production Engineering  
<sup>3</sup> s.schloesser@wzl.rwth-aachen.de

### Abstract (300-500 words)

The realization of radical innovations is a crucial success factor for manufacturing companies acting in an environment of increasing market dynamics. Heterogeneous customer requirements, which are increasingly changing in shorter cycles, become a major challenge for manufacturers of technical systems. Alongside these external circumstances, functional complexity of technical systems increases, which constitutes another major challenge in product development. As an answer to similar circumstances, the software industry introduced agile development methods in the early 1990s. The iterative development of functional product increments being shippable to potential customers at the end of a development phase has helped the industry to dynamically align the product to the customers' needs and to reduce the development time significantly. In this context, the concept of Minimum Viable Product (MVP) has gained particular attention. The MVP is a product, carrying just enough functionality to satisfy customers and to gather feedback for subsequent development with the least amount of effort and time. Taking the Lean Start Up initiative into account, the idea of MVP is generalized to be applicable to any type of new business or product. However, the aspect of selling incomplete products is commonly considered to be hardly applicable to technical systems. Nevertheless, the idea of iteratively validating major concerns and risks in product development applying the least amount of time and resources is equally agreed as a necessary step to tackle the challenges described above. Based on user stories for the product to be developed, major concerns and risks are deduced to be cleared out as early as possible with impacted stakeholders. Following the Design Thinking approach, Feasibility (Can we do this?), Desirability (Do they want this?) and Viability (Should we do this?) are types of uncertainty to be reduced as far as possible with each iteration. Due to increasingly interconnected technical systems, most of these subjects are very likely to impact several non-separable functions and disciplines within the system at once. Considering this along with the inherent fact of expensive and time-consuming realization of physical prototypes, the application of the MVP principle becomes a major challenge itself.

Therefore, this paper introduces a holistic framework to elaborate an appropriate methodology to support the planning and design of prototypes in agile product development of technical systems. Thereby, the methodology provides an approach to implement physical prototypes in

line with the *Minimum Viable Product* principle. As part of the overall methodology, the paper outlines a procedure to systematically structure design variables, which are meant to be specified and validated by a prototype with respect to particular concerns and risks under investigation. Therefore, the affected functions are systematically graduated and described by generic prototype features, capturing generic design elements such as the geometrical shape, the operating principle or the performance level which enables an initial evaluation whether a feature needs to be specified and captured within the physical realization. Based on this classification, a target-oriented design of physical prototypes is facilitated. Subsequent research will focus on further validating this procedure and on detailing the overall methodology, encompassing the procedure subject to this paper.

**Keywords:** *Product Design, Agile Product Development, Prototyping, Minimum Viable Product*

## **1 Introduction**

The realization of radical innovations in an environment of heterogeneous and dynamic customer requirements has evolved into a substantial challenge for manufacturers of technical systems (O'Connor et al., 2013). Since agile development principles have turned out to be very beneficial for software development under similar circumstances in the last decades, the adaption of these principles to the development of technical systems is currently being pursued in various industries (Schuh et al., 2016a). In this context, particularly the role of prototyping and the associated challenges are increasingly discussed in industrial practice as well as in scientific publications (Böhmer et al., 2017).

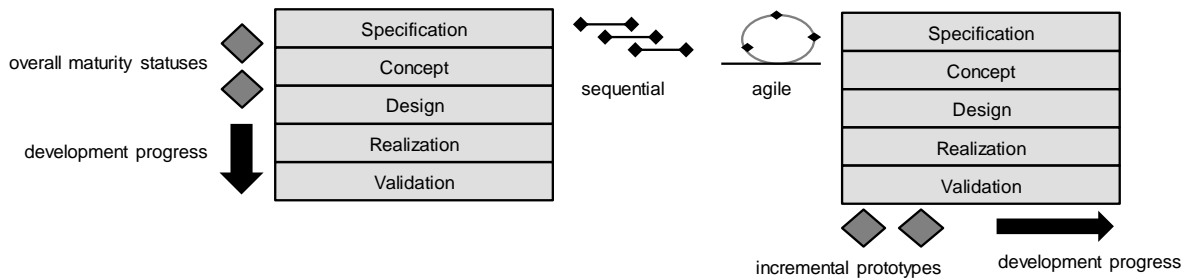
Therefore, this paper examines the characteristics of prototyping in agile development of technical systems while focusing concrete challenges in industrial practice (section 2). As the outlined subject has already provoked contributions from various fields of research, the state of the art is presented in section 3. A methodological approach to address the identified challenges and to equally close the identified gap in design literature is presented in section 4 before the paper closes with a conclusion and an outlook to future research (section 5).

## **2 Characteristics of prototyping in agile product development for technical systems**

The Agile Manifesto in 2001 initially introduced the principles of agile product development. Several paradigms and principles for a new approach in software development were manifested, which focus customer satisfaction through early and continuous delivery of valuable software rather than following a strict development plan (Beck et al. 2001). In this chapter, the processual characteristics of agile product development are outlined (section 2.1) before the specific role of prototypes in the context of adapting agile principles to the development of technical systems is introduced (section 2.2). Finally, section 2.3 emphasizes current challenges, which industrial practice is currently facing in the course of introducing agile principles.

### **2.1 Agile product development process**

The core processual difference between traditional development approaches for technical systems and the agile development approach is illustrated by figure 1. While traditional processes are characterized by a sequential tracking of overall maturity statuses, agile processes rather consider distinct incremental prototypes as key indicators for development progress (Schloesser et al. 2017).

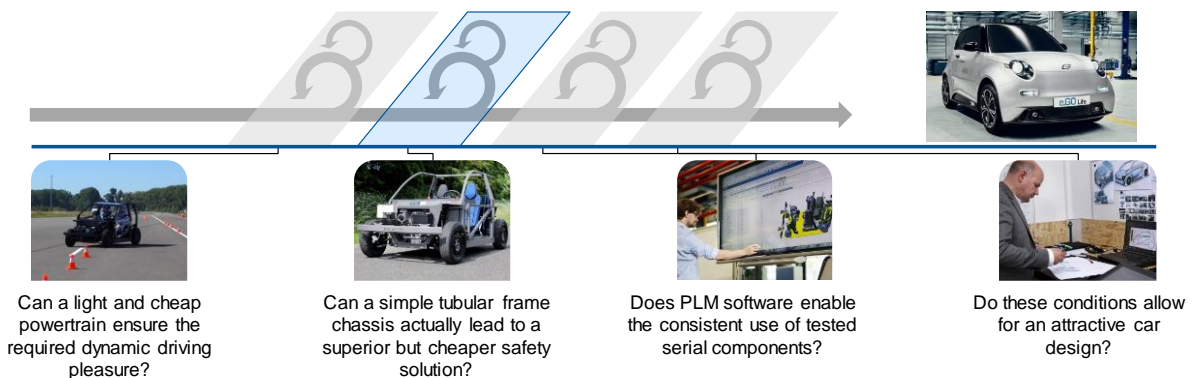


**Figure 1. Traditional development approach vs. agile development approach (Johnson, 2011; Schloesser et al., 2017)**

While various agile development frameworks are available today, *Scrum* has been the most commonly used one (Schwaber, 1997). Due to its popularity, *Scrum* is mostly taken as the basis to implement agile principles to the development of technical systems. *Scrum* divides the product development process into *Sprints*, which are iterative development cycles in fixed timeframes. Hereby *Sprints* target at the delivery of distinct product increments to be validated at the end of a *Sprint*. During *Sprint planning*, the product scope to be developed in a *Sprint* is defined. While in software development it is widely agreed to target at deployable software within any *Sprint*, agile development of technical systems rather focusses on validating tangible prototypes with external and internal stakeholders (Gabrysiak et al., 2010; Schuh et al. 2017). In this context, Schoeneberg emphasizes the major benefit of iterative prototyping in continuously ascertaining, specifying and validating design variables with customers (Schoeneberg, 2014). As a consequence, uncertainties, which are mainly caused by “absence of sufficient confidence in design variables that prevent design process participants to specify or predict a priori the system being designed - its behaviors and properties - in a deterministic and quantitative way” (Fernandes et al., 2013), are gradually eliminated.

## 2.2 The role of prototyping

The role of prototypes as it has historically grown in traditional stage-gate development processes, is widely considered to significantly change in agile development processes (Böhmer et al., 2017). Prototypes are no longer planned upfront to provide proof of development at pre-defined stages. In fact, prototypes are planned iteratively to explore major uncertainties (Kampker et al., 2016). These major concerns are captured by formulating explicit *Questions* (Cooper et al., 2016, Schuh et al. 2017). Consequently, *Questions* and the associated prototypes determine both the timeframe and the necessary activities within an agile development process (Schuh et al. 2017). Figure 2 exemplifies how this approach has characterized the development process of a small electric vehicle at RWTH Aachen University, Germany (e.GO Mobile AG, 2018).



**Figure 2. Agile development process based on *Questions* and associated prototypes [Schuh et al., 2017]**

According to this, *Questions* and associated prototypes determine development tasks and efforts in the development process. In contrast to that, development tasks and efforts as well as prototypes are in most companies mainly plan-driven and pre-defined in stage-gate development processes. Consequently, for most companies and the project leads in charge the iterative planning and design of appropriate prototypes is a rather novel task when adapting agile development principles to technical systems.

### 2.3 Resulting challenges in industrial practice

Due to the inherent fact, that functions of today's technical systems are commonly strongly interrelated, *Questions* are very likely to impact several non-separable functions and disciplines within the system at a time. While a detailed prototypical realization of every function is hardly possible, it is yet crucial to maintain the *Question's* functional context to a certain extend. This fact has been captured in a simplified illustration by *Spotify* Agile Coach Henrik Kniberg (figure 3).

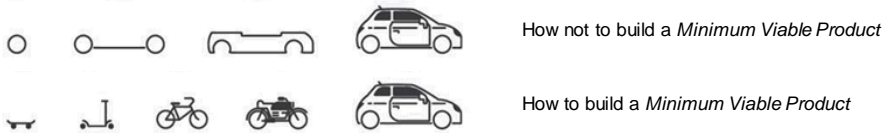


Figure 3. Minimum Viable Product according to Spotify (Kniberg, 2013)

While the upper path illustrates the approach of dividing functional blocks into separate prototypes, the bottom path follows the mentioned approach of maintaining the functional context of the desired product within a prototype. From a high-level perspective, the core functions of an individual passenger mobility solution such as *accelerating*, *steering*, *integrating passenger* are considered in all of the evolutionary steps while being individually abstracted to a certain extend. For example, working principles, the performance level or further design variables of individual functions are approximated (Wall et al., 1992) since the comprehensive exploration of a *Question* by all affected design variables is purely not possible and even not necessary in early prototypes.

Based on this exemplification, the general challenge in applying agile development processes to technical systems is illustrated on an abstract level by figure 4.

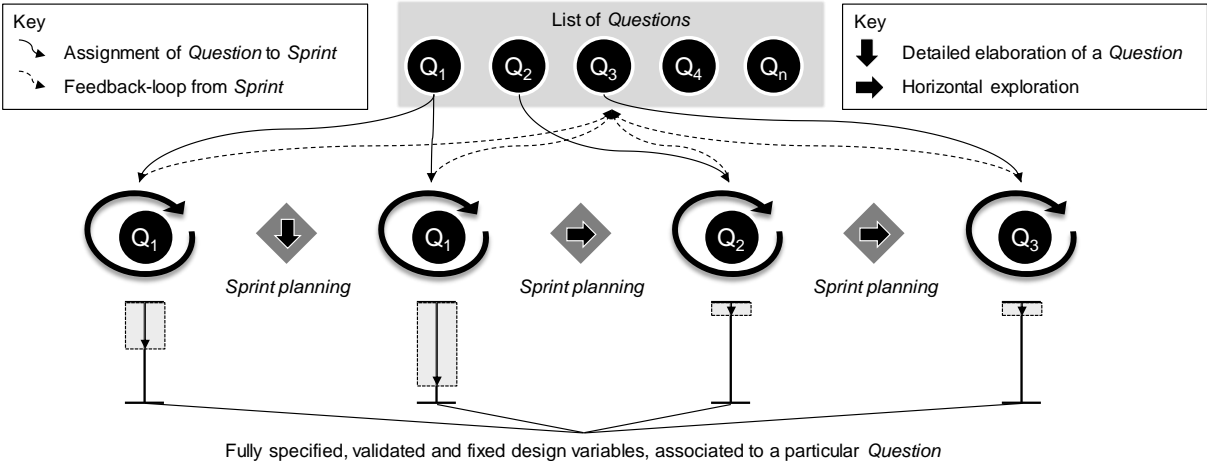


Figure 4. Sprint planning in agile development processes

Within the *Sprint planning* of any iteration a set of design variables has to be defined, which shall be subject to the elaboration of the particular *Sprint*. Following the *Minimum Viable Product* (Ries, 2011) approach, this set aims at exploring a particular *Question* in a way that

delivers reliable insights to major uncertainties with the least amount of effort in a given timeframe. At the same time, designers naturally strive to thoroughly solving problems (Terho et al., 2016), which is eventually required to successfully release technical systems (Ovesen, 2012). Accordingly, the defined set of design variables determines, whether an iteration focusses on the horizontal exploration of different *Questions* on system level or rather targets at the vertical elaboration of a particular *Question* and the associated design variables in detail. As illustrated in figure 4, the described problem recurs with any iteration of an agile development process and is substantial to efficiently and effectively apply agile development principles to technical systems.

Therefore, a methodology is developed to systematically support this recurring planning process. This paper outlines the overall methodological approach to address the illustrated challenge and introduces a procedure, facilitating the systematic framing and prioritization of design variables in order to design prototypes with respect to particular *Questions* according to the *Minimum Viable Product* principle. Based on this procedure, further models will be developed to systemize the *Sprint planning* process by a methodological decision support.

### **3 State of the art**

The general propagation of agile methods to the project management and control of technical systems development in combination with the rise of generative manufacturing technologies for rapid prototyping such as 3D-printing has recently drawn the attention of numerous publications to prototyping for technical systems in the context of agile product development. Moreover, prototyping is one of the most costly and time-consuming processes in design and in the same course it is widely considered to be increasingly important to align development activities to customer's volatile needs. Therefore, research has investigated the planning and design of prototypes extensively within the last decades.

To capture related work holistically, this section will cover the state of the art in planning and designing of prototypes for technical systems as well as relevant publications in the area of R&D project management and control. Moreover, related work, which considers prototype design as an integral part of the agile project management and control will be focused in this section.

The process of *planning and designing of prototypes for technical systems* has been explored by Albers et al. (Albers et al., 2016) who describe a general framework to structure validation activities in product development. Targeting at a description for prototypes, Exner et al. propose different types of prototypes from a material study through a functional prototype along the product development process (Exner et al., 2015). Similarly, Hilton et al. consolidate 20 criteria into a classification of prototypes based on a comprehensive review of existing literature. The classification comprises both physical characteristics as well as the intended use of prototypes (Hilton et al., 2015). Lim et al. describe the role of prototype twofold. First, prototypes serve as filters, capturing only qualities of interest while being as incomplete as possible. Secondly, prototypes manifest designers' ideas to make development tangible and facilitate iterative feedback loops. Both roles result in a description model, which consists of five filtering dimensions and three manifestation dimensions, describing the anatomy of prototypes (Lim et al., 2008). Based on a literature review, Christie et al. elaborate nine factors, being generally applicable to any prototyping endeavor. These factors comprise the general decisions to be taken when designing prototypes (Christie et al., 2012). In a similar way, Camburn et al. outline four generic decisions to be taken when targeting at reducing prototype cost and time. At the same time, it is emphasized to consider a prototype's fidelity, which predetermines the quantity and quality of achievable insights (Camburn et al., 2017). The fidelity of a prototype can be

modelled by means of an approach by Wall et al., who provide a framework of 13 constitutive attributes, describing a prototype's characteristics. The prototype's fidelity is modelled by a performance vector, which is defined by the set of attributes, for any component of the prototype. Each attribute is assigned a value from 0 to 1 representing the fidelity of a component to the most likely final design (Wall et al., 1992). Consolidating the core ideas of most of the above mentioned approaches, Tronvoll et al. provide six general factors, which lead to inherent trade-offs in prototyping design. According to this, iteration costs and time, approximation level, user level, result presentation and experiment flexibility have to be evaluated and prioritized to efficiently and effectively design prototypes (Tronvoll et al., 2017).

Since this paper targets at facilitating the systematic allocation of design variables to individual *Sprints* and the corresponding prototypes, approaches from the research area of *R&D project management and control* are equally taken into account as related work. In this context, Rauhut provides a methodology to structure and synchronize development tasks through systematic allocation of tasks to defined development cycles (Rauhut, 2011). A methodological approach of set-based design, facilitating a continuous reduction of the design space, is developed by Lüdtkke. This approach enables a time-discrete evaluation and choice of design solution alternatives and associated design variables (Lüdtkke, 2016). Agile-stage-gate processes are recently considered as promising approaches to structure development activities of technical products. Cooper et al. advice to enhance traditional stage-gate processes by agile processes whenever high uncertainties have to be managed and great need for customer involvement is required. At this, incremental product versions (*Protocepts*) promise to enable quick learning cycles and frequent customer interaction (Cooper et al., 2016). Following the approach of combining stage-gate processes and agile processes, Schuh et al. present a framework to evaluate the suitability of agile processes and conventional stage-gate processes. Based on distinct dimensions such as customer relevance, market & technology uncertainty as well as prototype manufacturability, particular product scopes in terms of functions and components are assigned to either one of both processual approaches (Schuh et al., 2016b).

Lastly, several approaches already consider *prototype design as an integral part of project management and control* in the context of agile development processes. Initial attention to rather prototype-based structured development processes has been gained by Lundahl (Lundahl, 2006). Based on the stage-gate process, presented in 2001 by Cooper (Cooper, 2001), the author seeks to strengthen customer involvement within the stages to continuously align development activities to the customers' needs. In this context, the term *Protocept* has mainly been constituted as being both a concept and rough prototype which is testable. An initial *Protocept* is suggested to be designed to create a business case and shall continuously evolve during the development process to eventually capture the full product definition (Lundahl, 2006). In a similar manner, Hoffmann et al. define five methodological types of prototypes, representing particular phases of a development process. Starting with a critical function prototype, capturing only most critical features from a customer's perspective, a final prototype is meant to eventually represent the products' features towards the customer as it was a real product (Hoffmann et al., 2016). While these types of prototypes remain on a methodological level, Böhmer et al. investigate the role of prototypes in an agile development process with respect to functionalities being carried out. Thereby, the observation of prototyping activities of different design teams revealed different strategies, that can be pursued when exploring a product's major uncertainties and iteratively achieving a fully specified product (Böhmer et al., 2017). Mainly focusing on the systematic reduction of uncertainties in an agile development process, Gartzzen et al. provide an approach to select a suitable prototype which is appropriate to address the focused uncertainties within a *Sprint* (Gartzzen et al., 2016).

The mentioned approaches focus on systemizing the utilization of prototypes as an integral part of the agile development of technical systems as well as this paper does. However, the state of the art should not neglect how the mentioned challenges are addressed in software development, which has widely been applying agile development processes for years. *Scrum* as the most commonly used development approach, foresees a dynamic product backlog capturing the current state of requirements at any time of the project. Based on the product backlog the team chooses elements, which can be transferred to a functional prototype within one *Sprint* (Schwaber, 1997). According to Pichler, the elements of the product backlog are always structured based on priorities, which can be set by different criteria such as additional value, reduced risk or required effort (Pichler, 2013). To facilitate a tangible description of prototypes and associated product backlog elements, Hochreuter et al. suggest a filter-fidelity model. This model comprises generic design parameters, enabling a filtered description of features being considered within a prototype (Hochreuter et al., 2013).

In summary, related work touches the outlined challenges in agile prototyping for technical systems and provides suitable links to systemize the framing and prioritization of design parameters in order to design iteration-compliant prototypes of technical systems.

## **4 Iteration-compliant prototyping in agile product development**

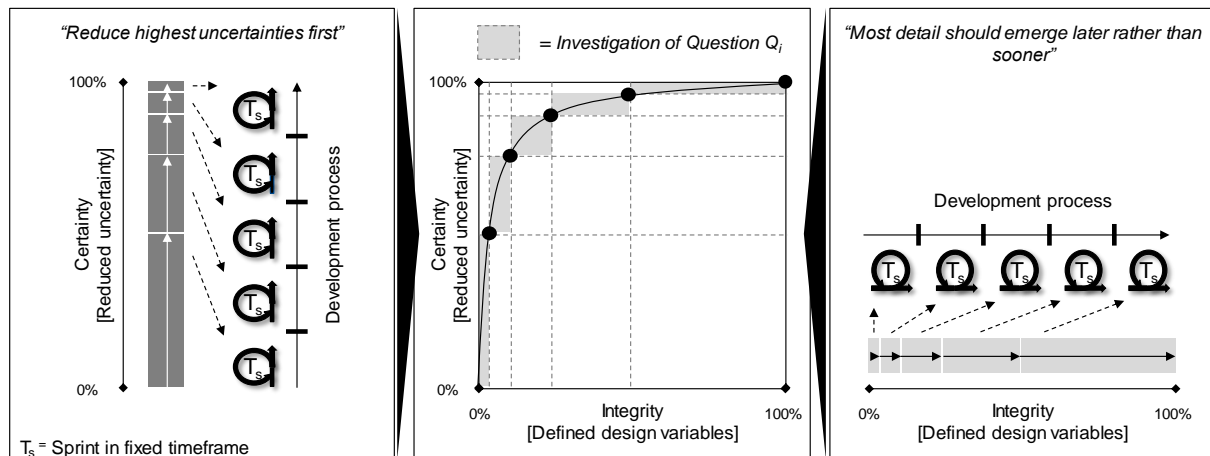
The exemplified challenge in industrial practice will be addressed by a systematic procedure, which is subject of this paper. Before this procedure is introduced in detail (section 4.3), the overall research framework is illustrated in section 4.2. In order to facilitate a clearer understanding of the required methodological steps, the underlying research hypothesis is outlined beforehand (section 4.1).

### **4.1 Research hypothesis**

Uncertainty in product development is inevitable due to the “absence of sufficient knowledge, definition or confidence in design variables that prevent design process participants to specify or predict a priori the system being designed – its behaviors and properties – in a deterministic and quantitative way” (Fernandes et al., 2013), especially in the early phases of product development processes. Accordingly, product development activities basically seek to define design variables in a way such that uncertainty is transferred into confidence about a product’s behaviors and properties, which are feasible, desirable and viable.

The main promise of introducing agile development processes to technical systems lies in better adjusting product development to the customers’ needs while requiring less development time compared to conventional development approaches (Schuh et al., 2017). Therefore, major uncertainties in terms of *Questions* are explored in the early phases of product development by incremental prototypes to gather, specify and validate main design variables through intensive customer involvement (Schoeneberg, 2014). In this context, the first *Sprints* are meant to reduce the highest uncertainties of the overall development project so that the absolute uncertainty reduction will gradually decrease through project completion (figure 5; left). At the same time it is commonly known that the early phase of agile development processes is usually characterized by vague requirements, so that specifying of most design variables is hardly possible within the first *Sprints*. Furthermore, an early definition of too many design variables might prohibit innovative solutions or causes corrective rework in the course of development (Lüdtke, 2016). In addition to that, the Agile Business Consortium manifests to let most details of development work emerge later rather than sooner within the core principles of agile development (Agile Business Consortium, 2014). However, any development of technical systems eventually requires integrity of all design variables to allow for marketable products

(Ovesen, 2012). The outlined evolution from a set of only few defined design variables to an eventual fully specified product is depicted in the right part of figure 5.



**Figure 5. Research hypothesis**

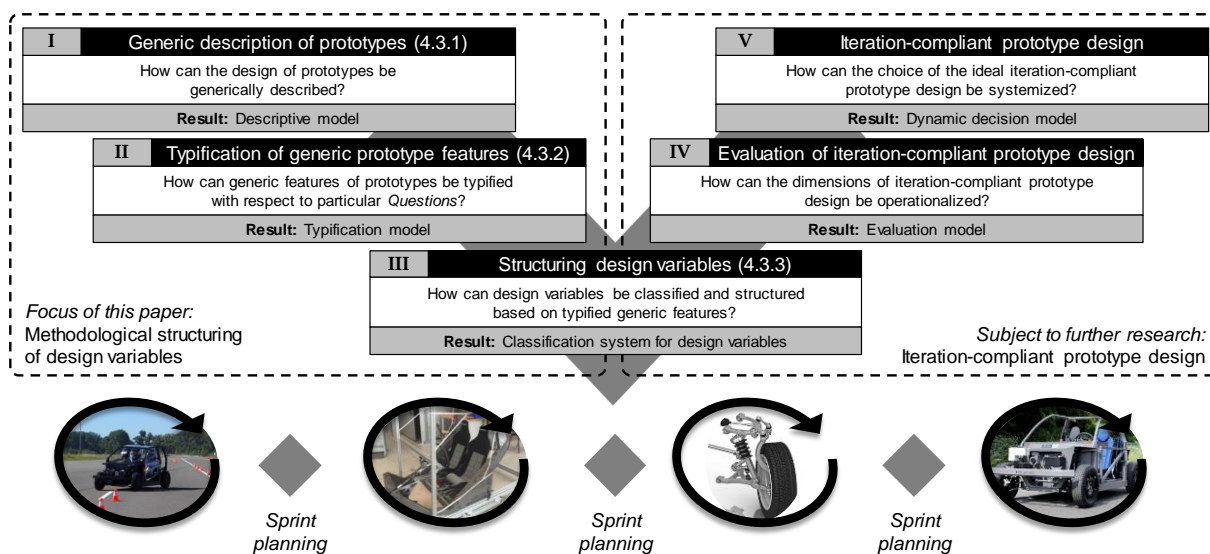
Combining both perspectives (figure 5; middle) outlines how an agile development process ideally evolves according to the paradigms described above.

As a consequence, the underlying research hypothesis of this work is summarized as follows: An iteration-compliant prototype contributes to an appropriate increase in certainty and accounts for eventual integrity of design variables according to the respective phase of the development process. Moreover, an iteration-compliant prototype’s realization in a fixed timeframe is systematically ensured.

Consequently, a methodology to support *Sprint planning* is required to enhance the iteration-compliant design of prototypes.

## 4.2 Research framework

Based on the research hypothesis a framework is developed, which addresses the elaboration of the methodology in five steps (figure 6).



**Figure 6. Elaboration of a methodology to support iteration-compliant prototype design**



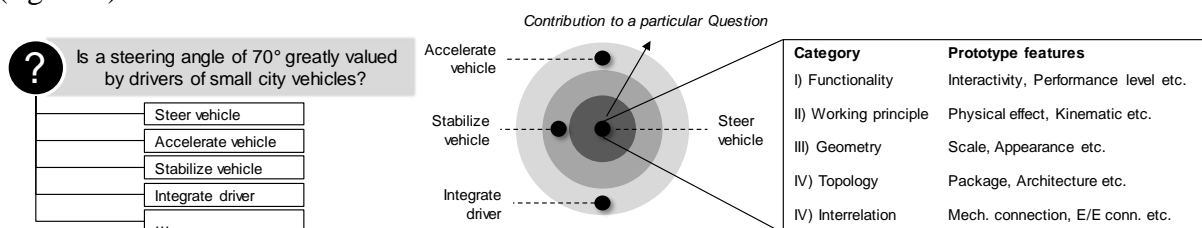
In a first step, a model to generically describe prototypes is developed in order to filter the relevant aspects of the overall development task, which are meant to be captured within a prototype. A second model targets at typifying generic prototype features with respect to particular *Questions*, which capture the major uncertainties of an overall development project (section 2.3). A third model is required to establish the link-up of the methodology with the concrete use case. For this reason, the set of individual design variables is linked with the typified features and is classified accordingly. As introduced in the research hypothesis, it is crucial to provide methodological support in iteration-compliant prototype design, which is facilitated by the fourth model. This model operationalizes the dimensions *certainty*, *integrity* and *time*, which have to be considered in iteration-compliant design according to figure 5. The choice of the most appropriate and iteration-compliant design variables to be assigned to the next prototype is subject to a dynamic decision model, being developed in a final step.

### 4.3 Structuring of design variables for iteration-compliant prototype design

As indicated in the overall framework (figure 6), this section introduces three partial models, yielding a procedure for systematically structuring of design variables in order to deduce iteration-compliant prototypes with respect to particular *Questions*.

#### 4.3.1 Generic description of prototypes

Referring to previous work from Schuh et al., particular *Questions* being explored in the agile development of technical systems typically affect a certain product scope, which can be represented by functions or components (Schuh et al., 2016b). As already illustrated by the example in section 2, prototypes of technical systems usually require the representation of a function's context to generate reliable insights about its properties and behaviors as part of the system. For this reason, it is firstly suggested to describe the functional product scope by using a shell model (figure 7). The shell model captures the affected functions as well as its individual contribution to a particular *Question*. Secondly, functions of technical systems can be described by five categories of design variables (Breunig, 2017). Based on a comprehensive review of publications in the area of *planning and designing of prototypes for technical systems* (section 3), the generic prototype features can be consolidated and are assigned to the categories (figure 7).



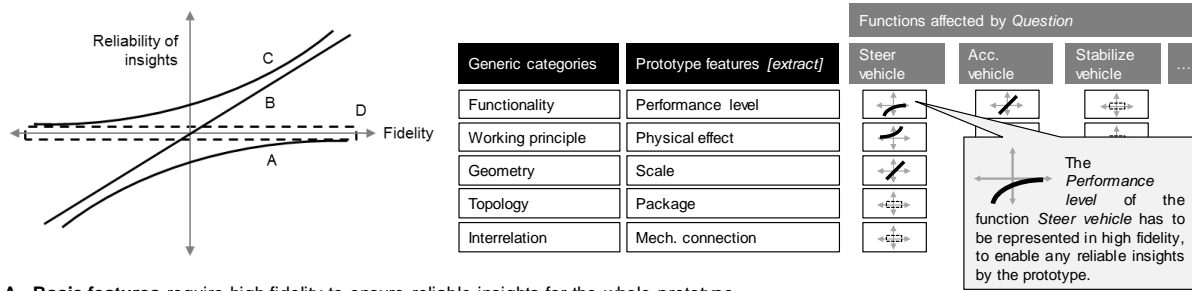
**Figure 7. Generic description of prototypes**

By means of this model, the filtered aspects of an overall development task which are subject to be captured within a prototype can be structured.

#### 4.3.2 Typification of generic prototype features

According to the *Minimum Viable Product* approach by Ries, a prototype should only carry those features which are necessarily required to gather reliable insights to a particular *Question* (Ries, 2011). In order to determine the relevance of generic features with respect to a particular *Question*, this model provides a methodological approach based on an adapted *Kano-principle* (Kano et al., 1984). As exemplified in figure 8, the model allows for a systematic typification

of prototype features with respect to a particular *Question*. Similar to the *Kano* model, four types are foreseen that prototype features can be assigned to and indicate the role of a feature.



**A - Basic features** require high fidelity to ensure reliable insights for the whole prototype.

**B - Performance features** enable extensive valid insights when detailed in high fidelity. Being represented in low fidelity, reliable insights for the whole prototype are jeopardized.

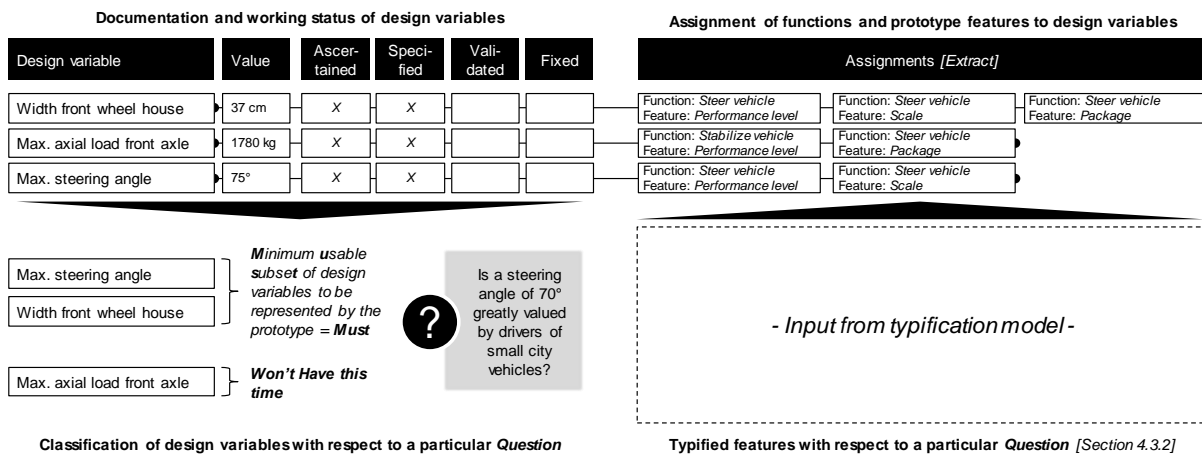
**C - Facultative features** enable extensive valid insights when detailed in high fidelity. Being represented in low fidelity, still enables reliable insights for the whole prototype.

**D - Irrelevant features** do not contribute to the generation of insights regardless of the fidelity.

**Figure 8. Typification of generic prototype features**

### 4.3.3 Structuring design variables

In order to provide usable results for concrete development projects, individual design variables of the product to be developed have to be embedded to the methodology. For that reason, the third model foresees a systematic procedure to document, initialize and dynamically structure design variables in agile development processes. As agile development processes explicitly take into account a volatile list of design variables, which might even barely exist in the early phase of development, the methodology has to be prepared to dynamically receive and process current states of design variables in any *Sprint planning*. Accordingly, a structure of documenting volatile design variables is exemplified (figure 9 top left). Once a design variable is ascertained, it is initialized by assigning functions and prototype features which are affected by the individual design variable (figure 9 top right). As illustrated, the model can be fed by prototype features, which are typified with respect to a particular *Question* (figure 9 bottom right). Based on this input, the list of design variables can automatically be structured according to the *Question* being examined. For that reason, MoSCoW (Agile Business Consortium, 2014) as an existing prioritization method for requirements in agile project management serves as the basis to systematically frame design variables (figure 10 bottom left).



**Figure 9. Structuring design variables**

Based on this model, a systematic framing and prioritization of design variables is facilitated, which enables to design prototypes according to the *Minimum Viable Product* principle.

## 5 Conclusion and further research

The role of prototypes when introducing agile principles to the development of technical systems has initially been outlined. Furthermore, the challenges in industrial practice have turned out to occur especially in the process of prototype design as part of the *Sprint planning*. At this, iteration-compliant prototype design has been proclaimed as a suitable concept, to ensure early reduction of major uncertainties and to eventually accomplish integrity of design variables, while permanently obeying the principle of delivering prototypes in fixed timeframes. In order to systematically support the recurring *Sprint planning* process, this paper introduced a holistic framework to elaborate an appropriate methodology. The groundwork for the methodology was presented in three models, guiding through the process of systematically structuring design variables with respect to particular *Questions* to be explored in *Sprints*. In the course of the initial application within the agile development of a small electric vehicle at e.GO Mobile AG, the general practicability has been approved. However, it has been noticed, that the definition of prototype features might have to be specified with respect to the context of an industry to enable an even stronger link to the particular use case.

For practitioners the overall methodological framework can serve as a basis to initially structure the process of *Sprint Planning* when adapting agile product development principles to technical systems. Furthermore, the presented procedure can already be applied to prioritize and structure development tasks in agile development processes. To the academic design community the framework serves as another building block in the overall initiative of exploring how agile principles can be adapted to technical systems development.

Future research will focus on further validating the presented models in industrial practice and enhancing individual steps by insights gained during application in *Sprint planning*. Based on the existing models, further models will be developed to complete the methodology as indicated in the overall research framework.

## 6 Bibliography

- Agile Business Consortium (2014) Agile Project Management Handbook.
- Albers, A., Behrendt, M., Klingler, S. & Matros, K. (2016) Verifikation und Validierung im Produktentstehungsprozess, in Lindemann, U. (2016) Handbuch Produktentwicklung, Hanser, München.
- Beck, K., Beedle, M, van Bennekum, A., Cockburn, Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mallor, S., Shwaber, K. & Sutherland, J. (2001) The Agile Manifesto, available from: <http://www.agilemanifesto.org/>. [12.03.2018]
- Böhmer, A. Hostettler, R., Richter, C., Lindemann, U., Conradt, J., Knoll, A. (2017) Towards Agile Product Development - The Role of Prototyping, Proceedings of the 21<sup>st</sup> International Conference on Engineering Design (ICED17), Vol. 4: Design Methods and Tools, Vancouver, Canada, 21.-25.08.2017.
- Breunig, S. (2017) Produktarchitekturgestaltung mechatronischer Baukastensysteme, Dissertation RWTH Aachen.
- Camburn, B., Viswanathan, V., Linsey, J.; Anderson, D., Jensen, D., Crawford, R., Otto, K. & Wood, K. (2017) Design prototyping methods, Design Science, vol. 3, e13.
- Christie, E. J., Jensen, D. D., Buckley, R. T. & Crawford, R. H. (2012) Prototyping strategies: Literature review and identification of critical variables, Proceedings of the 2012 American Society for Engineering Education Conference, pp. 1-22.

- Cooper (2001) *Winning At New Products: Accelerating The Process From Idea To Launch*, Basic Books.
- Cooper, R. G. & Sommer, A. F. (2016) From Experience: The Agile–Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity, *Journal of Product Innovation Management*, vol. 33, Issue 5, pp. 513-526.
- e.GO Mobile AG (2018) available from: <http://e-go-mobile.com/> [12.03.2018]
- Exner, K., Lindow, K., Stark, R., Angesleva, J., Bahr, B. & Nagy, E. (2015) A transdisciplinary perspective on prototyping, *IEEE International Conference on Engineering, Technology and Innovation/ International Technology Management Conference (ICE/ITMC)*, pp. 1-8.
- Fernandes, J., Henriques, E. & Silva, A. (2013) A Method for Managing Uncertainty Levels in Design Variables during Complex Product Development, in Aiguier, M., Boulanger, F., Krob, D. & Marchal, C. *Complex Systems Design & Management*, Springer International Publishing.
- Gabrysiak, G., Edelman, J., Giese, H., Seibel, A. (2010) How Tangible can Virtual Prototypes be?, *Proceedings of the 8th Design Thinking Research Symposium*, 2010, pp. 163-174.
- Gartzen, T., Brambring, F. & Basse, F. (2016) Target-oriented prototyping in highly iterative product development, *Procedia CIRP* 51, pp. 19-23.
- Hilton, E., Linsey, J. & Goodman, J. (2015) Understanding the prototyping strategies of experienced designers, *Proceedings of IEEE Frontiers in Education Conference (FIE)*, pp. 1-8.
- Hochreuter, T., Diefenbach, S. & Hassenzahl, M. (2013) Durch schnelles Scheitern zum Erfolg: Eine Frage des passenden Prototypen?, *Tagungsband UP13*, pp. 78-84.
- Hoffmann, C., Lennerts, S. & Schmitz, C. (2016) *Business Innovation: Das St. Galler Modell*, Gabler.
- Johnson, N. (2011) *Applying Agile To Hardware Development*, Xtreme EDA USA Corporation, available from <https://www.agilealliance.org/wp-content/uploads/2016/01/applying-agile-to-ic-development.pdf> [12.03.2018]
- Kampker, A., Förstmann, R., Ordnung, M., & Haunreiter, A. (2016) Prototypen im agilen Entwicklungsmanagement, *ATZ - Automobiltechnische Zeitschrift*, Ausgabe 7-8/2016, pp. 7-8.
- Kano, N., Seraku, N., Takahashi, F. & Tsuji, S. (1984) Attractive quality and must-be quality, *Journal of the Japanese Society for Quality Control (in Japanese)*. April, 1984, 14 (2): pp. 39–48.
- Kniberg, H. (2013) *How Spotify builds products*, available from: [https://www.dcc.fc.up.pt/~mcoimbra/lectures/SIM\\_1415/SIM\\_1415\\_T2.3\\_HowSpotifyBuildsProducts.pdf](https://www.dcc.fc.up.pt/~mcoimbra/lectures/SIM_1415/SIM_1415_T2.3_HowSpotifyBuildsProducts.pdf) [12.03.2018]
- Lim, Y-K., Stolterman, E. & Tenenberg, J. (2008) The anatomy of prototypes, *ACM Transactions on Computer-Human Interaction (TOCHI)*, Nr. 2, 15. pp. 7:1 – 7:27.
- Lundahl, D. (2006) *Holistic Product Development for New Product Success: Advancing Beyond Stage-Gate*, available from: [https://www.dcc.fc.up.pt/~mcoimbra/lectures/SIM\\_1415/SIM\\_1415\\_T2.3\\_HowSpotifyBuildsProducts.pdf](https://www.dcc.fc.up.pt/~mcoimbra/lectures/SIM_1415/SIM_1415_T2.3_HowSpotifyBuildsProducts.pdf) [12.03.2018]
- Lüdtke, B. (2017) *Lösungsraum-Steuerung in der Produktentwicklung*, Dissertation RWTH Aachen.
- O'Connor, G. C., Rice, M.P. (2013) A Comprehensive Model of Uncertainty Associated with Radical Innovation, *Journal of Product Innovation Management*, vol. 30, Issue S1, 2013 pp. 2-18.
- Ovesen, N. (2012) *The Challenges of Becoming Agile: Implementing and Conducting Scrum in Integrated Product Development*, Aalborg University.

- Pichler, R. (2013) Scrum: Agiles Projektmanagement erfolgreich einsetzen, Dpunkt, Heidelberg.
- Rauhut, M. (2011) Synchronisation von Entwicklungsprozessen durch Taktung, Dissertation RWTH Aachen.
- Ries, E. (2011) The Lean Startup - How today's entrepreneurs use continuous innovation to create radically successful businesses, Crown Publishing Group, New York.
- Schoeneberg, K.-P. (2014) Komplexitätsmanagement in Unternehmen, Springer Fachmedien, Wiesbaden.
- Schloesser, S., Riesener, M. & Schuh, G. (2017) Prototyping in highly-iterative product development for technical systems, Proceedings 7. WGP-Jahreskongress Aachen, 5.-6. Oktober 2017, pp. 149-157.
- Schuh, G., Rudolf, S., Riesener, M., Kantelberg, J. (2016a) Application of Highly-Iterative Product Development in Automotive and Manufacturing Industry, Proceedings of ISPIM Innovation Forum, 2016, pp. 1-13.
- Schuh, G., Riesener, M. & Diels, F. (2016b) Structuring Highly Iterative Product Development Projects by Using HIP-Indicators, Proceedings of the IEEM Conference, pp. 1171-1175.
- Schuh, G., Riesener, M., Ortlieb, C., Diels, F. & Schröder, S. (2017) Agile Produktentwicklung, in Brecher, C., Klocke, F., Schmitt, R. & Schuh, G. (2017) Internet of Production für agile Unternehmen, Tagungsband AWK Aachener Werkzeugmaschinen-Kolloquium 2017, 18. bis 19. Mai, pp. 29-51.
- Schwaber, K. (1997) SCRUM Development Process, in Sutherland, J., Casanave, C., Miller, J., Patel, P., Hollowell, G. Business Object Design and Implementation, Springer London, London, 1997, pp. 117–134.
- Terho, H., Suonsyrjä, S. & Systä, K. (2016) The Developers Dilemma: Perfect Product Development or Fast Business Validation?, Proceedings of the 17th International Conference, PROFES, 2016, pp. 571-579.
- Tronvoll, S., Elverum, C. & Welø, T. (2017) Prototype Experiments - Strategies and Trade-offs, Procedia CIRP, 60, pp. 554 - 559.
- Wall, M., Ulrich, K. & Flowers, W. (1992) Evaluating Prototyping Technologies for Product Design, Research in Engineering Design, vol. 3, pp. 163 - 177.