NordDesign 2022
August 16-19, 2022
Kgs. Lyngby, Denmark

Increasing Transparency for Novel and Existing Complex Systems Development - Modular Design in an Agile Context

Marc Zuefle, Christoph Rennpferdt, Dieter Krause

Institute of Product Development and Mechanical Engineering Design, Hamburg University of Technology
marc.zuefle@tuhh.de
christoph.rennpferdt@tuhh.de
krause@tuhh.de

Abstract

This paper addresses the question of how Modular Design methods can support the development of complex products. The goal of selected Modular Design methods is to improve transparency in the development of modular systems to be able to structure these systems in a way that adds value to the observed company and to derive optimized product architectures from the conducted procedures. The problem addressed here is that existing methods can develop increasingly less transparency within the development of complex products, since the extent of these products rises increasingly. This increases the complexity of the work steps and scope of work, which results in a reduction of the immediate transparency in observing the underlying product architecture. To address this pain points, two selected methods in the context of Modular Design are exemplary adapted to agile approaches and principles. These adaptations are applied to two underlying case studies, each of which has to do with the development of complex products in various aspects. In the case studies, different process perspectives are depicted, which offer both a development process (further development of an existing system) of a laser cutting machine tool of a mechanical engineering company, as well as a development project (new design of a system with a high degree of novel development) of a smart industrial valve of a medium-sized engineering manufacturer with, in addition, an unknown proportion of new development as an investigation environment. The results show the integration of agile aspects in selected Modular Design methods as a successful support in generating transparency in the development of complex products and systems, in different process perspectives. At the same time, the research also shows that adaptations to the studied methods are necessary, such as continuous and centralized concept documentation, to apply them in an agile context.

Keywords: modular design, complex systems, product architecture, integrated product development, agile adaption

1 Introduction

As a result of the increasing demand for customized products and constant technological progress, manufacturing companies are forced to constantly expand and improve their range of products and services and to rethink and adapt their business models. This leads to ever shorter development cycles and, at the same time, to ever more complex systems and products, as market requirements have to be responded to more flexibly and intensively. In the medium to long term, this development leads to a strong increase in internal complexity within the company and thus to higher costs in the development and manufacture of such systems. (Blecker & Abdelkafi, 2006; ElMaraghy et al., 2012; Krause et al., 2014)

An increasing diversity of offerings often leads to an increasing number of product components and processes within a company. This causes direct costs, such as higher administrative expenses and additional storage space requirements, but also increases complexity costs, such as higher error rates due to decreasing transparency within the company. (Krause et al., 2014; Ripperda & Krause, 2017)

To counteract this trend, the literature refers to the development of modular product architectures. These allow different products to be configured from a set of existing components, rather than developing a new product for each customer (Krause et al., 2014). Another way to increase transparency in the development of such complex systems and products is to map these products in a digital twin (Laukotka et al., 2021). Model-based simulations of product architectures also enable supported viewing of various aspects of a complex systems (Seiler et al., 2019). However, the development of modular product architectures, as well as the mapping in a digital twin and the model-based support in the context of increasingly complex products, is also becoming more and more demanding. Examples of this are the development of Product-Service Systems (Isaksson et al., 2009; Reim et al., 2015; Rennpferdt & Krause, 2020) or cyber-physical systems (Porter & Heppelmann, 2014; Tomiyama et al., 2019) in which various additional aspects have to be considered. The question that arises in this context is how transparency in the development of complex systems can be enabled again and thus the successful development of such systems can be supported.

2 Research Background

A literature review by Yassine (Yassine, 2021) serves as the basis for the consideration of the research background. As mentioned in the introduction of this paper, products are becoming increasingly complex and mastering the development of these complex systems and products requires an adaptation of the underlying methods and approaches.

In the literature review, Yassine performs an integrative analysis of different theories, models, and tools that can be used to support the design of complex products. In doing so, Yassine distinguishes in three domains: *product*, *process*, and *people*, as well as their *common intersections*. (Yassine, 2021)

Due to the limitations of the paper, only the three overarching domains will be discussed in more detail below.

2.1 Product domain and Modular Architecture Design

In the *product domain*, the product architecture is defined. Here, the decomposition of the product architecture and a modular product architecture play a crucial role in handling the development of complex products (Yassine, 2021). Based on this, tools such as the *Design Structure Matrix* (DSM) (Steward, 1981), the *Modular Function Deployment* (MFD) (Erixon, 1998) or *Mechanical Electrical Software Architecture (MESA)* (Askhøj et al., 2021) can be mentioned for Modular Design. In addition methods are e.g., the *Design for Variety Method*

(Kipp & Krause, 2008), the *Life Phase Modularization* (LPM) (Blees et al., 2010) as well as *METUS* (Göpfert, 1998). Furthermore, approaches to this domain are exemplary the *Integrated PKT-Approach* (Krause et al., 2014) or the *Product Family Master Plan* (Simpson et al., 2012). The *Integrated PKT-Approach* enables a technical-functional and product-strategic view of the underlying product architecture. This enables a more holistic view and reduction of variant-induced complexity in product architectures. The core methods of the *Integrated PKT-Approach* are the *Design for Variety* (DfV) method according to Kipp and Krause (2008) and *Life Phase Modularization* (LPM) according to Blees et al. (2010). The *DfV* serves as preceding the *LPM*, but the *LPM* can also be applied independently. Figure 1 shows the two methods, including the method steps and their sequential character. Afterwards a short introduction in these methods is conducted.

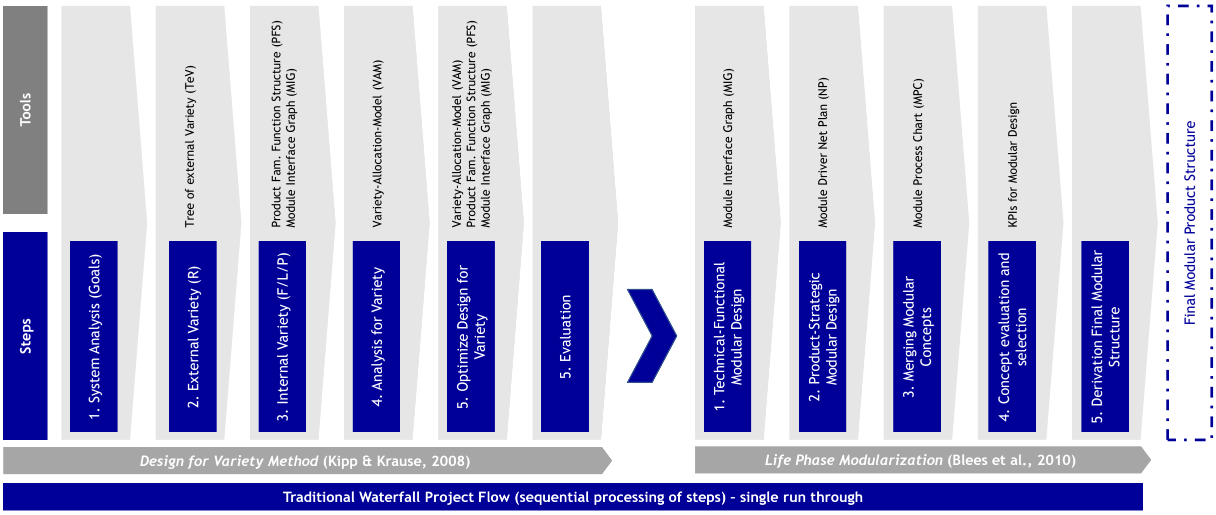

Figure 1. traditional sequential process of *Design for Variety Method* (Kipp & Krause, 2008) and *Life Phase Modularization* (Blees et al., 2010)

The method *DfV* by Kipp maps the external variety of a portfolio to the internal component variety via different analysis and synthesis steps. Thereby the external variety is mapped by the customer relevant product requirements (R) and the internal variety by functions (F), logics (L) and physical components (P). This categorization is somewhat similar to the classification from Systems Engineering (*RFLP-Approach* (Eigner et al., 2014)), but in *DfV* the focus is on the variety of elements. The goal of the method is to reduce and optimize the dependencies of internal component variety on relevant product requirements. An excerpt can be taken from Figure 1. (Kipp & Krause, 2008)

LPM by Blees harmonizes module cuts across all life phases involved so that the most uniform module cut possible can be mapped along the product life cycle. In the process, various life phase-specific module drivers are applied to the product architecture. The subsequent harminization results then in synergies in purchasing, maintenance, production and other life phases. (Blees et al., 2010)

2.2 Process Domain and Agile Product Development

The *process domain* refers to how different development projects are managed in the design process. This is partly about how a project is set up, including the process, requirements and results. As a proxy for this, three exemplary approaches emerge from research. *Agile product development* (Schmidt et al., 2018), such as the *Agile System Design Approach* by Albers et al. (2019) as an example, *Continuous Integration and Continuous Deployment*, such as by Galabba (2019), and *DevOps*, such as by Ebert et al. (2016) and Fitzgerald and Stol (2017). All these

approaches have their origin in software development but are more and more integrated into the development of mechatronic systems.
The *Agile Manifesto*, in which the four agile values and the twelve agile principles are laid down, serves as the basis for agile frameworks and techniques. The values include, for example, that the *interaction of individuals* should be given higher priority than the processes and tools that are used. An example of an agile principle is the *acceptance of change*. The rest of the values and principles can be found in the Agile Manifesto. (Beck et al., 2001)

2.3 People & Organization Domain

Finally, the *domain of people* addresses the organization of the communication pattern. Based on e.g., the *Mirroring Hypothesis* can be named. In the *Mirroring Hypothesis* it is concerned that the organizational structures adapt themselves in the course of the time to those of the product architecture (Colfer & Baldwin, 2016). In conclusion, the analysis shows that different approaches are used in different domains for the handling of complex products and the transparency required for this. Taking the analyzed results further, the product and people domains can be addressed by developing modular product families. The referred *Mirroring Hypothesis* by Colfer and Baldwin (2016) supports the clustering of *people* and *product*, as the organization adapts to the product architecture.
The *process* and *people* domains can exemplarily be covered by methods and approaches of agile product development with Continuous Integration and Continuous Deployment and DevOps, since these support the process of developing and testing complex products in e.g., several iteration steps.

2.4 Research Hypothesis

As a result of the analysis, the hypothesis arises:

Agile adaption support or complements the development of modular product architectures in handling the development of increasingly complex systems and products.

An analysis on agile adaptions in the development of modular product architectures, conducted on databases as Scopus and Web-of-Science, reveals a focus of publications on manufacturing and agility in the production processes. The analysis of the abstracts of the publications indicates, that there is hardly any reference to the development of modular product architectures. Further indications going in a similar direction can be taken from Birk et al. (2021) and Zuefle et al. (2022). As a conclusion, this hypothesis has to be examined by means of practical examples, which is conducted in the following.

3 Structure of Paper

From the literature review, it emerged that agile practices and Modular Design methodologies have not yet been extensively considered, but according to Yassine's review, can have a critical advantage in handling complex products. Derived from this, this paper addresses the hypothesis that an adaptation of Modular Design in terms of principles and values of agile practices can increase transparency and improve the development of complex products.
In order to examine the hypothesis, two case studies are considered which, in relation to Yassine's classification, allow two different views of the domain process. One is the novel project development and the other is the continuous development of existing systems. These case studies are described in Sections 4.1 and 4.2. Section 5.1 then introduces the discussed Modular Design methods and afterwards in Section 5.2 the values and principles of the *Agile Manifesto* as the basis for the agile approach are presented as fudamentals. Subsequently, the

discussed selection of Modular Design methods is combined in Section 5 with the presented values and principles to create an adaptation of the considered methods. To validate the adaptation, the presented case studies are used, in which the adaptation was applied. This results in findings for a successful adaptation, as well as still open points of consideration, which will be dealt with further in the discussion and the outlook.

4 Introduction of considered case studies

To be able to consider the points from the introduction, two case studies serve as examination objects. In this context, the various considerations are intended to provide a more comprehensive representation of the Process Domain introduced in Section 2. One case study looks at a *process-based development* (enhancement of an existing system) and the other at a *project-based development* (development project with a high degree of novelty). Both case studies deal with the development of complex systems, in which the transparency decreases for different reasons and is to be regenerated by suitable method applications.

4.1 Agile development of existing machine tools

The first case study in this publication relates to a project in an agile development environment for machine tools at a German manufacturing company. The objective of this project is to analyze and evolve the existing modular kit by using the *Integrated PKT-Approach* (Krause et al., 2014). The primary aspects are the structuring of customer functions offered, as well as their impact on the architecture in the modular kit and the integration of software and other development disciplines, such as electronics, fluidics and optics, in the methodical design of modular product architectures. This should enable the modular kit to respond more efficiently to customer requirements without having an excessively high impact on the developed architecture. The collaborative design of the product architecture, based on different development disciplines, leads to an increasing complexity in communication, modular design, process procedures and maintainability.

In order to be able to handle this complexity in these products and the modular kit, an agile development process has been established in the research & development environment of the case study. Semi-annual development processes for machine releases lead to continuous testing, adaptation, integration and deployment of the product versions. The development process itself is divided into sprints based on the agile framework *SCRUM* and includes defined milestones for interfaces and design changes.

To ensure that the project goals can be achieved in the context of the research environment, it is necessary to integrate the methods and tools used from the *Integrated PKT Approach* into the agile development process and the agile activities carried out within it, enabling them to be adapted accordingly.

4.2 Development of novel smart industrial valves

The second case study is a development project in the field of industrial valves. The objective of the project is to match the existing high variety of pressure reducing valves on the market with a newly developed modular kit. The *Integrated PKT-Approach* is used to design the modular kit in such a way that the internal variety of components is minimized as far as possible. An additional objective of the project is to enhance the existing pressure reducing valves with innovative sensor technology, thus enabling them for Industry 4.0 applications. These smart valves should record process data, such as pressure and temperature, and transmit them to the plant management system. The characteristic feature of the pressure reducing valves in question is that they are self-regulating valves that function without an external energy and signal supply.

The sensor technology being developed, therefore, needs to be self-sufficient and generate the required energy through energy harvesting.
The challenges in the project are the two partly contradictory objectives. On the one hand, the internal product variety should be reduced, and a modular kit should be implemented that covers the existing external product variety. On the other hand, an innovative sensor concept is to be developed to record and forward data what increases the internal variety. In contrast to the existing internal variety, the requirements for the new sensor technology are not completely defined at the beginning of the project. Therefore, the development of the sensor technology is performed iteratively. While the internal and external variety are captured and analyzed according to the *Design for Variety* method of the *Integrated PKT-Approach*, concepts for the measurement tasks, the energy harvesting and the data communication were developed in parallel. The goal of these initial concept developments was to determine more precisely which energy harvesting principles provide which amounts of energy and which parameters can be measured and how they can be sent. The results of the initial concept development of the sensor technology were aligned with the results of the analysis of internal and external variety and considered in the next step of *Design for Variety* method. In parallel, the concepts were further developed and concretized. Figure 2 shows the procedure described.

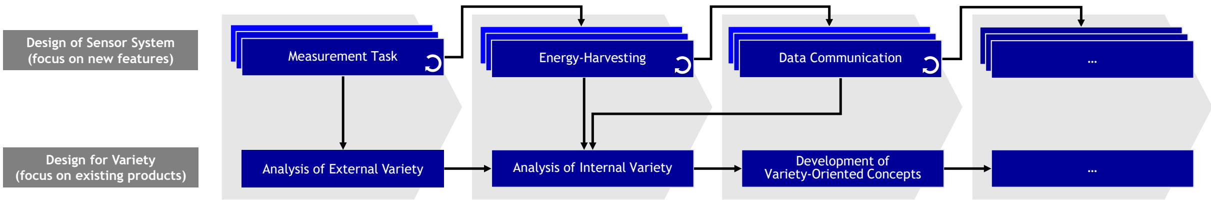

Figure 2. Schematic representation of the development process, redesign of the existing products (left) and iterative development of new product features (right) with the information exchange

5 Fundamentals of considered methods and agile aspects

Since a selection of modular design methods is to be adapted to an agile application in this work, the corresponding agile aspects from the process and people domain which are used for the adaptation are briefly discussed. For this purpose, the agile aspects, such as values and principles, are placed in a method development context, which is discussed in more detail. Subsequently, the findings are integrated and analyzed in the adaptation of the presented methods from the product and process domain.

5.1 Analysis of potential by agile considerations

As already discussed in the research background, agile methods and approaches offer a way to address the process and people domains in complex systems and products. The literature research conducted shows that there are differences in the approaches to agile development, which were dealt with more intensively for this paper. Due to the page limitations, the selection mentioned in the research background and the insights gained from it will not be discussed in more detail. For more extensive information, reference is made to e.g., Heimicke et al. (Heimicke et al.) among others. For this paper, the focus is primarily on a selection of agile values and principles by *Agile Manifesto* (Beck et al., 2001), as basis for agile approaches, that could be implemented in the two case studies and hence were the basis for further consideration. The values and principles were elaborated in the context of method development so that the meaning and objective of these values and principles could be translated into more focused descriptions for the analysis of methods in product development. This was necessary because the meaning of these values and principles is very general and there is a gap in the understanding of these aspects in method development. The focused interpretation of the values (updated to method requirements) and principles (updated to method characteristics) can be seen in Figure 3. These aspects were elaborated with 7 method developers in a three-step process including several workshops.

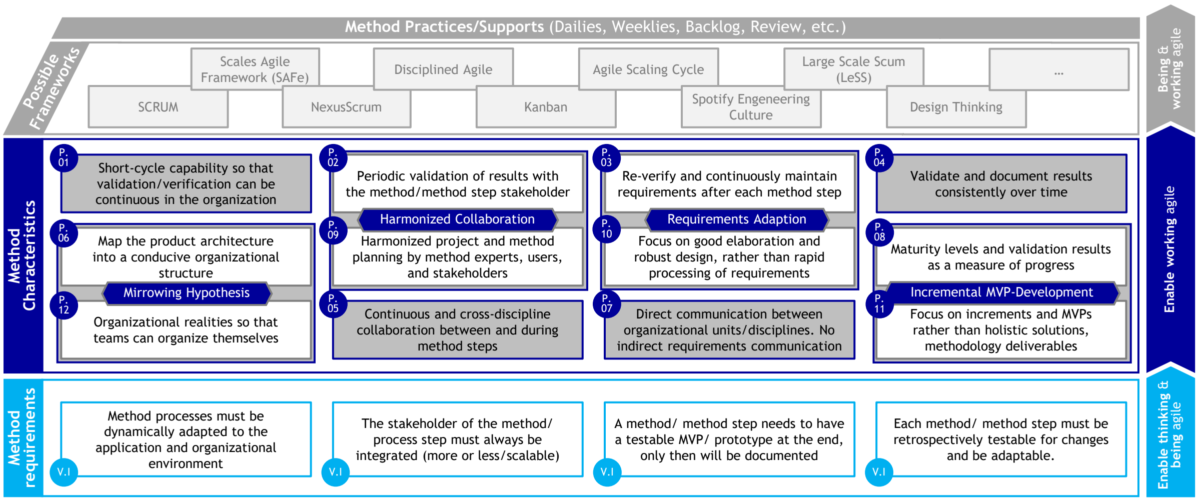

Figure 3. Adaption of agile values (bottom level) and principles (second level) on method development inspired by *Agile Manifesto* (Beck et al., 2001) extended by frameworks and practices (top levels).

The requirements and characteristics in Figure 3 are extended to include possible frameworks and agile practices. However, the focus of this elaboration is on the first two levels. The above definitions and classifications will be used for further consideration in this paper.

5.2 Agile Adaption of Modular Design Methods

The two case studies presented in Section 4 deal with the development of complex systems and products or with the development of products whose novelty or analogous constraints make the system/product complex. Modular product architectures are used or developed for these systems to manage the complexity of them and to use internal company potentials. For the further improvement of transparency and consistency of requirements, the methods and approaches for developing modular product architectures are extended to include a selection of the presented agile principles and values as updated method requirements and characteristics from Figure 3.

In the traditional application of the Modular Design (MD) methods presented in the case studies is sequential, as is depicted in Figure 1. Thus, first the product architecture is designed and adapted according to variety, so that the internal variety is optimized or optimally reduced, while the external variety remains the same (*DfV*). Then, in the *LPM*, a harmonization of the module sections is aimed at by integrating purchasing, production, sales, service, after-sales and other company-specific life phases. In this way, a common understanding of modules can be developed, taking into account different module drivers. (Krause et al., 2014)

In the context of complex products, such as an entire laser cutting machine or a new product with sensor components previously unknown to the manufacturing company, this sequential procedure leads to a very high effort in documentation, case and scenario considerations, as well as many uncertain parameters complicating the development.

By considering characteristics P.08, P.11 (both *Incremental MVP-Development* (Fig. 3), P.01 and P.12 („P“ stands for allocated agile Principle) an iterative adaption of the presented methods can support those principles in shortening the development cycles and increasing the traceability of continuous improvement. In addition, further adaptions have to be made to enable characteristics P.08 and P.11. Every iteration of the presented methods have to result in a functional increment, which can be validated and expended in following iterations. This can

be done by using Definitions-of-Done or similar for each method-iteration. In addition, by adjusting aspects from agile framework SCRUM time boxing processes support creating increments in this consideration (Zimmermann et al., 2021). To adapt characteristics P.03 and P.10 (both *Requirements Adaption* (Fig. 3)) to the presented methods, they have to be used in an intuitive and easy way. Thus both methods work with visualizations and simplified steps, this can be assumed as given (Krause et al., 2014). Characteristics P.02 and P.09 (both *Harmonized Collaboration* (Fig. 3)) are related to the stakeholders and the communication of results and ideas. By integrating product management and sales into *DfV* by Kipp and different life phases into the *LPM* by Blees, different Stakeholders can be integrated into the processes as early as possible and also validation procedures can be conducted in early stages. Characteristics P.06 and P.12 (*Mirrowing Hypothesis* (Fig. 3)) are about the self-organization of development teams as well as the self-dependent employees and can be adapted to the presented methods as well. Due to the fact, that the methods consider modelling a system these systems also can be defined as a subsystem. By limiting the observed system to a size of a development team, these teams can work self-organized and self-dependent on a specific scope. For validating and integration the increment into the supersystem the team has to constantly consider the requirements and characteristic P.11. Here for example, interfaces can be named. Due to the fact that the focus of an iteration is on a specific subsystem or subproblem, their interfaces have to be definied and documented, so that the increment can expand the previous functional product partials.
In conclusion, the methods discussed were placed in a subsystem and iterative context with respect to the organization of development teams and events, in which partial aspects of the overall product are processed in each cycle. This can be exemplary seen in Figure 4. Instead of analysing and developing a final product, the focus is set on a Minimal Viable Product (MVP) by focusing on subproblems or subsystems with a prioritized limited scope. This is defined by a Definition of Done which depicts the "Goals" of each iteration (also see Fig. 4 upper left and then each start of an iteration).

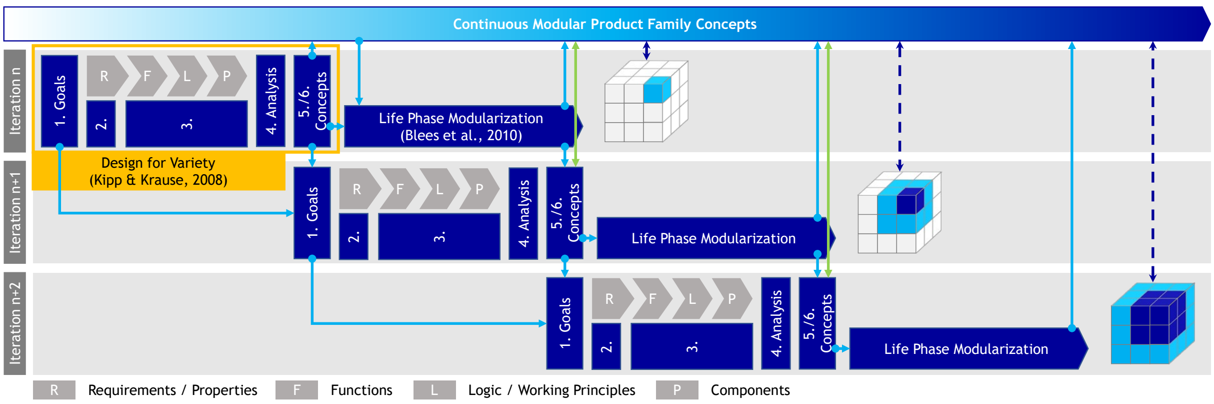

Figure 4. Extract Modular design in an agile context by continuous validation through iterations and a continuous documentation of all results and concepts in an overall memory. Compare to Figure 1.

For example, the Minimal Viable Product (MVP) is a converted industrial valve that has a mounted energy sensor wired to a computer standing beside. On the other hand, the focus can also be placed on a subsystem in the complex product to analyze and design a very specific functionality. In this way, the subsystems concerned can be considered and the peripheral subsystems added successively. These two options lead to the understanding that the designed system reaches different degrees of fulfillment each iteration in a specific scope. This is also depicted in Figure 4 by a 3x3x3 cube which illustrates the progress of the designed system.

In addition, for each step the validation and the testing of the requirements put into the process has to be performed. In an iterative procedure this can be done more frequently and

more specific on the investigated system. In Figure 4 this is illustrated in two ways. First the quality and validation degree of the developed system is depicted in the cube by different strengths of blue. This visualizes that due to iterative and incremental development, the results generated in iteration t-1 can be confirmed by the results from the subsequent iteration while building on the previous results. Furthermore, the design results and adapted requirements based on each iteration can be integrated in a *continuous Modular Product Family Concept* which exists parallel to all iterative executions of the methods. This enables a consistent documentation of all previous results and steps for further development and overall validation. Without this continuous memory of concepts, requirements and further sepcifications, as e.g., interface definitions, the agile adaption could not work in full amount.
In conclusion it can be shown, that the presented methods can be adapted to agile principles. For this, a few modifications must be done. For example, there has to be an continuous memory for documenting all increments and iterations. In addition, the consideration of external influences on subsystems has to be illustrated and mentioned.

6 Validation

Various aspects of the case studies presented can be used to validate the elaborated results.
In the process case study of the laser cutting machine, this can be described and analyzed using the example of a new derivative from the existing modular kit. This derivation should fulfill new customer-relevant properties and requirements, but internally be designed to be as variant-oriented as possible. Due to the novelty of the product's functionalities, the effects on the product architecture were not foreseeable at the beginning. To increase the transparency of the changes and effects on the product architecture, the *Design for Variety* and the *Life Phase Modularization* were applied iteratively in cross-functional teams and stakeholders. Figure 5 shows the iterations of the *Design for Variety*. It is well recognizable how the work with the representation becomes more concrete via the loops and leads to increasing transparency of the product architecture. In more detail, the understanding of the product architecture develops from a rough sense of which modules are affected to a detailed understanding of how the new functionalities affect components, processes, but also circuits due to joint workshops.
Building on the individual iteration results (increments), further life phases were involved. Using the *LPM*, it was possible to validate the impressions of the requirements in the visualization shown in Figure 5 and then to harmonize the common understanding. An exemplary result of the iterative *LPM* is that the product architecture could be harmonized with other life phases, such as sales and production, continuously and as early as possible. This makes it feasible to implement adjustments in the production process as well as in the product configuration at an early stage.

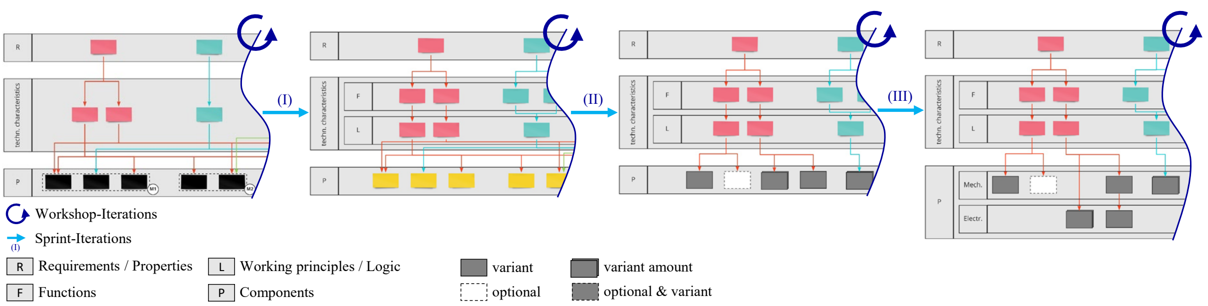

Figure 1. Abstracted representation of the Design for Variety iterations on an interactive online whiteboard.

By applying agile principles and methods of modular design, the development of the new product variety could be developed more quickly than previous projects. The early validation and adaptation, as well as the focus on subsystems, increased transparency already in the first iterations.
In the project case study of the valves, the challenge is that at the beginning of the development it is not clear how the customer-relevant feature "measure data" can be realized, because the amount of energy available is unknown. Therefore, the initial step is to analyze which working principle can be used to generate energy in the context of the valves. Based on this, tests on the test rig are used to evaluate which working principle provides which amount of energy, thus validating the preliminary considerations. The result of this iteration is a MVP. Using these results, the customer-relevant property "measure data" can then be more specifically defined, and in the next iteration, e.g., suitable sensor technology for measuring data can be developed, taking into account the amount of energy available.

7 Discussion

The context presented here, in which Modular Design can be thought agile, brings advantages to its application. Nevertheless, the context here does not consider the holistic agile mindset, values and princliples, as well as methods and tools. In this paper, the focus is on increasing transparency by adapting to a selection of agile princippels, as mentioned in Section 4.4. People as an integral part of product engineering are not explicitly addressed in this paper but must always be considered as well. For the analysis and conception of an agile modular design, the Integrated PKT-Approach was highly suitable in this case, because both it has various independent and scalable methods as a method construction kit, and it is already a part of the case studies. However, the Integrated PKT-Approach is not the only approach that deals with modular design. Other approaches must be considered separately within this context.
In conclusion, the analysis and findings presented here provide an interesting and usable construct that supports agile aspects and principles in the development of modular architectures and confirm, that agile aspects can lead to increasing transparency in modular design for complex systems.

8 Outlook

For the context described here to be further developed and concretized, further aspects and principles of agile product engineering and similar approaches must be investigated and integrated. The case studies continue to offer a good research environment for validating the concepts. In addition, the team composition and the collaboration of different stakeholders must be examined more closely in the further course of the research. The formation of cross-functional teams for early validation and adaptation creates challenges that were not directly apparent before. In addition, a framework of the process, needs to be explored in more detail, into which the methods considered fit.

Acknowledgements

Parts of this work are based on the project MODERATED (contract number 02K20K502) belonging to the program "KMU-innovativ: Produktionsforschung" funded by the German Federal Ministry of Education and Research.
The authors would like to thank all involved parties for their funding and support.

References

Albers, A., Heimicke, J., Spadinger, M., Reiß, N., Breitschuh, J., Richter, T., Bursac, N., & Marthaler, F. (2019). *Eine Systematik zur situationsadäquaten*

Mechatroniksystementwicklung durch ASD - Agile Systems Design. https://doi.org/10.5445/IR/1000091847

Askhøj, C., Christensen, C. K. F., & Mortensen, N. H. (2021). Cross domain modularization tool: Mechanics, electronics, and software. *Concurrent Engineering*, 1063293X2110003. https://doi.org/10.1177/1063293X211000331

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. http://www.agilemanifesto.org/

Birk, C., Zuefle, M., Albers, A., Bursac, N., & Krause, D. (2021). INTERDISCIPLINARY SYSTEM ARCHITECTURES IN AGILE MODULAR DEVELOPMENT IN THE PRODUCT GENERATION DEVELOPMENT MODEL USING THE EXAMPLE OF A MACHINE TOOL MANUFACTURER. *Proceedings of the Design Society*, *1*, 1897–1906. https://doi.org/10.1017/pds.2021.451

Blecker, T., & Abdelkafi, N. (2006). Complexity and variety in mass customization systems: analysis and recommendations. *Management Decision*, *44*(7), 908–929. https://doi.org/10.1108/00251740610680596

Blees, C., Jonas, H., & Krause, D. (2010). Development of Modular Product Families. In D. C. Wynn (Ed.), *Managing complexity by modelling dependencies: Proceedings of the 12th International DSM Conference Cambridge, UK, 22 - 23 July 2010*. Hanser.

Colfer, L. J., & Baldwin, C. Y. (2016). The mirroring hypothesis: theory, evidence, and exceptions. *Industrial and Corporate Change*, *25*(5), 709–738. https://doi.org/10.1093/icc/dtw027

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *IEEE Software*, *33*(3), 94–100. https://doi.org/10.1109/MS.2016.68

Eigner, M., Dickopf, T., Apostolov, H., Schaefer, P., Faißt, K.-G., & Keßler, A. (2014). System Lifecycle Management: Initial Approach for a Sustainable Product Development Process Based on Methods of Model Based Systems Engineering. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-45937-9_29

ElMaraghy, W., ElMaraghy, H., Tomiyama, T., & Monostori, L. (2012). Complexity in engineering design and manufacturing. *CIRP Annals*, *61*(2), 793–814. https://doi.org/10.1016/j.cirp.2012.05.001

Erixon, G. (1998). *Modular function deployment: A method for product modularisation*. Zugl.: Stockholm, Kungl. Tekn. Högsk., Diss., 1998. The Royal Inst. of Technology Dept. of Manufacturing Systems Assembly Systems Division.

Fitzgerald, B., & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, *123*, 176–189. https://doi.org/10.1016/j.jss.2015.06.063

Gallaba, K. (2019). Improving the Robustness and Efficiency of Continuous Integration and Deployment. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 619–623). IEEE. https://doi.org/10.1109/ICSME.2019.00099

Göpfert, J. (1998). Modulare Produktentwicklung. In N. Franke & C.-F. von Braun (Eds.), *Innovationsforschung und Technologiemanagement* (pp. 139–151). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-48173-4_12

Heimicke, J., Bramato, L., Schoeck, M., Müller, J., & Albers, A. Framework for Introducing Agility into Development Processes of Producing Companies. In *ISPIM Connects Valencia 2021 - Reconnect, Rediscover & Reimagine.*

Isaksson, O., Larsson, T. C., & Rönnbäck, A. Ö. (2009). Development of product-service systems: challenges and opportunities for the manufacturing firm. *Journal of Engineering Design*, *20*(4), 329–348. https://doi.org/10.1080/09544820903152663

Kipp, T. E., & Krause, D. (2008). Design FOR VARIETY - EFFICIENT SUPPORT FOR DESIGN ENGINEERS. In *10th International Design Conference Design 2008, May 19 - 22, 2008, Cavtat - Dubrovnik, Croatia: Proceedings.*

Krause, D., Beckmann, G., Eilmus, S., Gebhardt, N., Jonas, H., & Rettberg, R. (2014). Integrated Development of Modular Product Families: A Methods Toolkit. *Advances in Product Family and Product Platform Design.* (pp. 245–269). Springer New York. https://doi.org/10.1007/978-1-4614-7937-6_10

Laukotka, F., Hanna, M., & Krause, D. (2021). Digital twins of product families in aviation based on an MBSE-assisted approach. *Procedia CIRP*, *100*, 684–689. https://doi.org/10.1016/j.procir.2021.05.144

Porter, M. E., & Heppelmann, J. E. (2014). How Smart, Connected Products Are Transforming Competition. *Harvard Business Review* (11), 64–88. https://hbr.org/2014/11/how-smart-connected-products-are-transforming-competition

Reim, W., Parida, V., & Örtqvist, D. (2015). Product–Service Systems (PSS) business models and tactics – a systematic literature review. *Journal of Cleaner Production*, *97*, 61–75. https://doi.org/10.1016/j.jclepro.2014.07.003

Rennpferdt, C., & Krause, D. (2020). Towards a Framework for the Design of variety-oriented Product-Service Systems. *Proceedings of the Design Society: DESIGN Conference*, *1*, 1345–1354. https://doi.org/10.1017/dsd.2020.108

Ripperda, S., & Krause, D. (2017). Cost Effects of Modular Product Family Structures: Methods and Quantification of Impacts to Support Decision Making. *Journal of Mechanical Design*, *139*(2), Article 021103. https://doi.org/10.1115/1.4035430

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

Seiler, F. M., Greve, E., & Krause, D. (2019). Development of a Configure-to-Order-Based Process for the Implementation of Modular Product Architectures: A Case Study, 2971–2980. https://doi.org/10.1017/dsi.2019.304

Simpson, T. W., Bobuk, A., Slingerland, L. A., Brennan, S., Logan, D., & Reichard, K. (2012). From user requirements to commonality specifications: an integrated approach to product family design. *Research in Engineering Design*, *23*(2), 141–153. https://doi.org/10.1007/s00163-011-0119-4

Steward, D. V. (1981). The design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, *EM-28*(3), 71–74. https://doi.org/10.1109/TEM.1981.6448589

Tomiyama, T., Lutters, E., Stark, R., & Abramovici, M. (2019). Development capabilities for smart products. *CIRP Annals*, *68*(2), 727–750. https://doi.org/10.1016/j.cirp.2019.05.010

Yassine, A. A. (2021). Managing the Development of Complex Product Systems: An Integrative Literature Review. *IEEE Transactions on Engineering Management*, *68*(6), 1619–1636. https://doi.org/10.1109/TEM.2019.2929660

Zimmermann, V., Heimicke, J., Schnurr, T., Bursac, N., & Albers, A. Minimum Viable Products in Mechatronic System Engineering: Approach for Early and Continuous Validation. In *R&D Management Conference 2021 - Innovation in an Era of Disruption.* (Original work published 2021)

Zuefle, M., Muschik, S., Bursac, N., & Krause, D. (2022). COPING ASYNCHRONOUS MODULAR PRODUCT DESIGN BY MODELLING A SYSTEMS-IN-SYSTEM. *17TH INTERNATIONAL DESIGN CONFERENCE.* Advance online publication. https://doi.org/10.1017/pds.2022.258