Methodology for Determining Agile Product Scopes in Development Projects

G. Schuh, C. Dölle, F. Diels, M. Kuhn
Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Aachen, Germany

Abstract—**In today's fast-paced and unpredictable business environment, manufacturing companies are exposed to increasingly dynamic market conditions and volatilely changing customer expectations. Therefore, a complete definition of requirement specifications is often not possible, as customer needs in early development phases may strongly differ from needs in advanced phases. One way to encounter these challenges is the application of agile methods from software industry to the development of physical products. However, companies face high barriers during the implementation of agile methods in physical product development.**

This paper introduces a methodology to determine the application suitability of agile methods for highly interconnected physical products by using Agile Indicators. Agile Indicators are metrics that support the decision-making process for or against agile development methods at the level of individual product scopes. In order to do so, a classification of different types of development questions derived from uncertain requirements is presented. Subsequently, product scopes are determined in accordance with the existing development questions. In a next step, the derived product scopes are evaluated with Agile Indicators and the interdependencies between product scopes are analyzed. Based on this analysis a final selection of product scopes for agile development can be conducted.

I. Introduction

For manufacturing companies it becomes increasingly difficult to delight customers with innovative products that create a competitive advantage. Not only product complexity increases continuously, but also development projects themselves become more complex due to the involvement of different specialist departments such as hardware, software and electronics. Moreover, companies face increasing difficulties in fully identifying relevant customer expectations as well as quickly responding to changing requirements [1]. This leads to a lack of information during a product development project and in consequence to a level of uncertainty regarding product requirements. Some requirements even remain unknown until advanced stages of the development project. In order to operate successfully in today's competition, the ability to generate innovations quickly and to launch new products with a high level of efficiency becomes more and more important [2].

Generally, companies encounter these challenges with sequential plan-driven development approaches like waterfall processes or the Stage-Gate process by Cooper [3]. Development results are validated at concisely defined milestones in the development project, which is why deviations between development results and customer expectations can only be revised retrospectively. Research indicates that an increasing number of companies start to experiment with agile development approaches. Methods like Scrum, Kanban or Design Thinking, which originated in the software industry, are now being adapted by manufacturing companies of mechatronic products [4]. Agile development approaches combine a set of principles and methods in which solutions are generated iteratively in multiple development cycles, that reduce time-to-market as well as development expenses and lead to higher project success rates [5]. In contrast to plan-driven development approaches, agile approaches continuously adjust the development effort to volatile customer expectations or changes in technical or regulatory requirements and validate development results with various prototypes, so-called Minimum Viable Products (MVP).

Although, initial studies have confirmed the applicability of agile approaches to the development of physical products, a simple transfer is not purposeful for various reasons. On the one hand, there is a lack of theoretical principles and methodologies that apply the characteristics of agile development approaches to the restrictions of highly interconnected physical products. On the other hand, it is neither feasible nor appropriate to develop physical prototypes after at the end of each development iteration, as is customary in software development. Therefore, manufacturing companies tend to combine agile approaches with existing development structures in a hybrid development process [6]. Approximately two-thirds of companies that adapt agile development approaches use a hybrid or selective development process [7], which, according to Cooper, can be of considerable benefit. Hybrid development processes can lead to a better responsiveness to changing requirements, consider the voice-of-customer much more effectively than traditional plan-driven processes and reduce development cycle times [8]. However, a systematic identification and differentiation between agile and plan-driven development in a hybrid development process has not been sufficiently investigated in current scientific literature.

A promising approach to structure and coordinate a hybrid development process is the differentiation between agile and plan-driven product scopes on the component level, based on an evaluation with Agile Indicators [9]. It is reasonable to specifically identify product scopes that depend on incomplete and

978-1-890843-37-3 ©2018 PICMET

volatile requirements and are suitable for agile development. For this reason, certain product scopes should be developed either in an agile or plan-driven way. Consequently, interdependencies between product scopes with different development recommendations may occur and have to be analyzed. Some product scopes may contain components that simultaneously are part of another product scope with a different development recommendation. In order to create a hybrid development process, that is as efficient as possible a final decision model for determining whether a product scope should be developed in an agile or plan-driven way, must be established.

This paper builds upon a series of publications of the authors in the field of hybrid product development for physical products and the evaluation of individual product scopes with help of Agile Indicators [9] [10] [11] [12]. Based on latest findings, strategies to deal with interconnected product scopes with different development recommendations are outlined. The methodology presented in this paper helps to optimize the use of agile methods in the context of physical products and supports the structuring of a hybrid development process.

II. State of the Art

In the following, existing research on hybrid product development approaches will be presented. Special focus is placed on approaches that address the characteristics of physical product development.

Cooper & Sommer recently published the "Agile-Stage-Gate Hybrid Model" as a new development approach for manufacturing companies that combines conventional plan-driven or phase-based development with agile methods such as Scrum [6]. The combination of those two development practices aims at greater flexibility, shorter time-to-market and better responsiveness to changing requirements. The authors emphasize that additional adjustments of common agile development methods have to be made due to special characteristics of physical products. For instance, as the physical increment at the end of a development sprint is not divisible, manufacturing companies have to redefine the completion of a sprint. The authors define a "protocept" - a composite word of prototype and concept - as something that can be shown to the customer to obtain feedback. Accordingly, an implementation of agile methods and the quick generation of prototypes is reasonable, if requirements are unknown or incomplete. Although the potential of a hybrid development is outlined, a concrete methodology for distinguishing between agile and plan-driven product scopes is not constituted.

Conforto et al. analyze the combination of agile methods and key elements of Stage-Gate processes in the context of technology-based companies [13]. Central element of their research is a hybrid framework named "Iterative and Visual Project Management Method", which is characterized by various planning and controlling levels. The first level is defined as the conventional Stage-Gate process with certain milestones and development phases, followed by the second level, which is related to iterative development respectively Scrum. The third level visualizes relevant activities on a weekly or daily basis on a planning whiteboard. After the relevant information is gathered and implemented in a project management software, it can be assessed in a performance indicator system. However, a specific distinction between agile and plan-driven development scopes is not taken into account.

Feldhusen et al. describe a methodology for determining the optimal level of agility in a hybrid development for physical products [14]. The aim is to iteratively build targeted prototypes, which only validate the main requirements in the first step and consequently reduce uncertainty in the overall product development. Therefore, the authors develop an Agile Indicator, which is a metric composed of several influencing factors such as team size or completeness of requirements. The Agile Indicator is assessed on a scale between -1 for a strictly plan-driven development and 1 for a purely agile development. The Agile Indicator is determined at the beginning of a development project, but is not a constant value and may change due to new boundary conditions or other discontinuities. The methodology represents a significant contribution to the theory of hybrid product development of physical products, but focuses primarily on the overall project level. No distinction is made between individual product scopes, so that interdependencies between individual product scopes are not taken into account.

Klein analyzes the systematic integration of agile methods into the development of mechatronic systems [15]. "Agile Engineering" is defined as a hybrid development process, which describes the development and production of potentially deliverable product increments with the help of agile methods. In order to take the special characteristics of physical product development into account, the author defines several influencing factors such as the type of product, organizational structure, team size or the degree of modularization. Depending on its characteristic, each influencing factor is assessed on an agility scale between 1 and 4. A high evaluation corresponds to a high suitability of agile methods for the investigated development project. Although, the developed methodology considers interdependencies of components through the influencing factor "degree of modularization", it still focuses on the overall project without a differentiation between individual product scopes.

Eklund et al. identify a set of key practices for scaling agile development projects beyond individual teams in the context of mechatronic product development [16]. Based on a data collection from several case studies, the authors have identified three feedback loops that run in parallel in a hybrid product development. The inner loop, similar to a team sprint in the Scrum methodology, is carried out by individual teams and typically consists of the development of modules or components. The middle loop aims to coordinate the work of multiple teams and integrates hardware, software and mechanics. The third loop is used to obtain customer feedback at longer intervals. Each loop is applied in a different context, delivers different results and has a different cycle time. Moreover, the authors stress the issue of interdependencies between individual product scopes, as they emphasize that a high modularity is an asset in a hybrid product development. Furthermore, it is suggested to shift functional complexity from hardware to software, if possible, in order to dissolve phyiscal interdependencies. Nevertheless, the authors are not refering to an evaluation of individual product scopes.

In summary, the related approaches show the importance and topicality of agile methods for the development of physical products. However, none of the presented approaches in the field of hybrid development adresses an evaluation of individual product scopes and analyzes interdependencies between product scopes entirely. The question of how to differentiate between product scopes that should preferably be developed with either agile or plan-driven development methods remains unanswered. Therefore, the following methodology adresses the existing research deficit. The aim is to differentiate between product scopes that should be developed in an agile or plan-driven way based on an evaluation with Agile Indicators and the analysis of interdependencies between individual product scopes.

III. METHODOLOGY

The methodology is divided into three sections (see Fig. 1). In order to manage and reduce prevailing uncertainty in a development project, individual product scopes, which base on insufficiently or incompletely described requirements, have to be determined. Thus, the first step of the methodology is the deduction of development questions resulting from inadequately described requirements and the systematic clustering of product scopes from these requirements. Subsequently, these individual product scopes are evaluated regarding their suitability for agile development. This evaluation takes place by means of Agile Indicators, which enable a market and company-specific, reproducible assessment. Finally, a selection of agile product scopes in a hybrid product development is conducted. This last section particularly deals with the consideration of interdependencies between individual product scopes. As this paper is the fifth in a recent series of publications in the field of hybrid product development by the authors, the focus will be on the third section, while the first two sections of the methodology are kept shorter with reference to previous publications [9] [10] [11] [12].

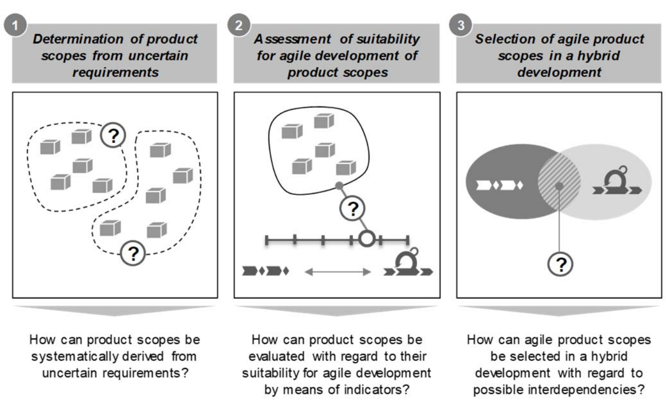


Fig. 1 Overview of the methodology

A. *Determination of product scopes from uncertain requirements*

At the beginning of a product development project, a systematic requirements management is used to gather, analyze and document relevant product requirements from a wide variety of sources. The result of this initial phase is a list of requirements, the so-called specification sheet [17]. The specification sheet describes in detail what various stakeholders expect from a product. It is important to note that a specification sheet is characterized by changing boundary conditions such as new insights of customer groups, extended technology potential or resource availabilities. These potential changes in the description of requirements lead to a certain incompleteness of the specification sheet and consequently to uncertainty in a product development project. By applying agile development methods, existing uncertainties can be systematically reduced through continuous validation of intermediate development results and short-cycled integration of customer feedback. This has been proven to significantly increase efficiency and effectiveness in product development [18].

In order to systematically identify existing uncertainties in the specification sheet and to derive individual product scopes, which can be developed with agile methods, so-called development questions have to be deduced. In the context of this methodology, a development question is any kind of question with regard to the product development that can be derived from one or more requirements which are not fully determined. It is possible to derive several development questions from one requirement, as well as several requirements can lead to the same development question. Accordingly, "m" development questions can be derived from "n" requirements with $n > m$, $n = m$ or $n < m$. With regard to the degree of detail and abstraction the following classification of types of development questions can be derived on the basis of different classes of correlating prototypes [19]: conceptual questions, design questions, functional questions and technical questions (see Fig. 2). A conceptual development question deals with the visual appearance and proportions of a model. The specific geometric shape and surface is linked to a design development question. While a functional development question addresses the validation of individual functions on module level, a technical development question serves to verify the overall function of the product.

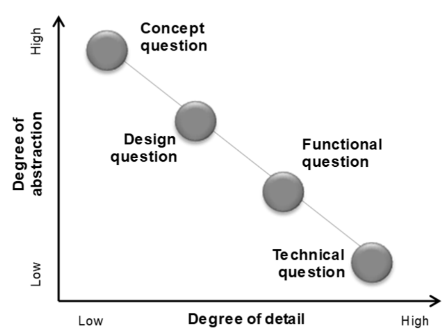


Fig. 2 Types of development questions

In order to purposefully respond to individual development questions and thus reduce uncertainties in the overall development project, various types of targeted prototypes have to be derived [20]. Therefore, individual product scopes consisting of various components, which specifically address a prevailing development question, need to be identified. The Design Structure Matrix (DSM) is a suitable method to identify and visualize possible interactions in technical systems. In the context of the present methodology, this is accomplished by a description of development questions, product functions and product components, which consequently lead to a derivation of appropriate product scopes for an agile development. For this purpose, the DSM are extended by Domain Mapping Matrices (DMM), which allow a cross-domain analysis of interactions between development questions, functions and components (see Fig. 3).

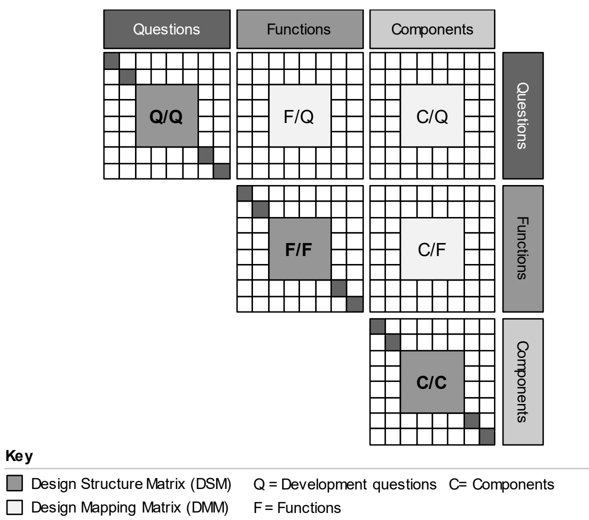


Fig. 3 Underlying Design Structure Matrices (DSM) and Design Mapping Matrices (DMM) for the present methodolgy

The structure consists of three DSM and three DMM. The first DSM (Q/Q) combines development questions, as they cannot be considered independently. Especially in view of limited time and financial resources of a development project, the analysis of similarities of development questions and the targeted response of bundled prevailing questions is essential. The second DSM (F/F) enables the hierarchization and aggregation of functions and the third DSM (C/C) displays interdependencies between components. The three DMM are used for the assignment of functions to development questions (F/Q), i. e. which functions are required to answer a prevailing question, the assignment of components to development questions (C/Q), i. e. which components directly address a question, and the assignment of components to functions (C/F). The latter DMM provides information on the necessary components for the implementation of a function and thus represents the correlation between the functional and physical description of the product. Reference functions and product structures, which are based on predecessor or reference products and established in most manufacturing companies, help to conduct this methodological step.

In order to be able to derive product scopes within the component DSM (C/C), an analysis using a cluster algorithm is required. In general, a cluster represents a structural subarea of an overall system in which the contained elements are stronger linked to each other than to external elements. In the context of this paper, the definition of a cluster corresponds to a product scope resulting from a development question. In the component DSM (C/C) individual components form a product scope if the interdependencies between components are as close as possible to the diagonal of the matrix and thus form a definable cluster. This identification can be achieved by rearranging the matrix columns and rows using an algorithm. Different algorithms for solving this optimization problem are described in various scientific literature. Since the focus of this paper is a methodology for determining product scopes in context of agile development, in-depth description on analyzing clusters by means of algorithms can be found in current scientific literature [21] [22] [23].

Two important characteristics of the first step of the present methodology should be emphasized. Firstly, the derivation of product scopes is not a static process at the beginning of an agile development project, but should take place in an iterative process due to changes in the initial situation of a development project. Secondly, it is not possible to derive completely autonomous product scopes within complex technical products. The decomposition of a product into individual product scopes using a cluster algorithm is a mathematical optimization problem, in which interdependencies usually exist [24]. Therefore, it is necessary to search for an “adequate good” decomposition and to systematically deal with the remaining interdependencies. This is conducted in the third step of the present methodology.

The result of this first section is a set of product scopes that are aligned with prevailing development questions. Each individual product scope aims at answering at least one development question. In the next step of the methodology, product scopes are evaluated with regard to their suitability for agile development.

B. *Assessment of suitability for agile development of product scopes*

After relevant product scopes, which specifically address prevailing development questions, have been identified, this section focuses on determining the suitability of an agile development approach with help of Agile Indicators. For this purpose, internal and external Agile Indicators are derived, based on internal and external influencing factors of the project. Subsequently, these are aggregated into an overall Agile Indicator. The Agile Indicator provides a recommendation for the decision-making process on whether a product scope should be developed with plan-driven or agile methods [11].

Since the derivation of internal and external influencing factors has been described in detail in recent publications of the authors [9] [10], only a brief explanation of these factors is given here in order to illustrate the overall context. Five internal and three external influencing factors for an agile development were identified through an extensive literature research and numerous expert interviews.

From an internal point of view the *solution space* of a product development, the *resources*, which affect and restrict the capacity for action, the *technology ability* of a company, the *prototype manufacturability* and the *corporate culture* are crucial influencing factors for an agile development [9]. As alternative solutions can change during the development of a product, a large solution space indicates a preferred suitability for agile development. This is due to the fact, that agile methods iteratively reduce the solution space through the validation of interim results. Moreover, the availability of qualified employees, corporate knowledge, existing machinery or the available project budget are decisive resources for or against the use of agile methods. Furthermore, a company's adaptability to emerging technologies is a key characteristic. If a technical requirement changes over time and leads to a shift in its technological realization, the selection between an agile or plan-driven approach is influenced. The prototype manufacturability constitutes another crucial influencing factor, as different kinds of prototypes help to purposefully respond to individual development questions and subsequently reduce uncertainties in a development project. Finally, the corporate culture is a decisive internal influencing factor for an agile development. Current research investigates the cultural readiness of companies planning to adopt agile methods through assessing cultural aspects [31]. Management support, self-organizing teams, agile specific training, project allocations and the acceptance of agile roles are some of the most important cultural aspects.

From an external point of view the *market relevance*, the *market accuracy* and the *market volatility* are essential influencing factors for an agile development [10]. The market relevance can be illustrated by the Kano-Model. It divides customer requirements into irrelevant requirements, basic requirements, performance requirements and delighters. Agile development is suitable when a product requirement is of great interest for the customer, as it iteratively validates and aligns product specifications with customer expectations. The market accuracy describes the preciseness or correctness of recorded product requirements. Due to a limited customer's abstractive capability in terms of formulating desired product characteristics and functionalities, product requirements are often imprecise at the beginning of a development project. The less accurate the requirements, the more sensible an agile development approach is. Finally, the market volatility indicates the amount of uncertainty regarding the extent of changes and variations of product requirements along the development project due to dynamic market conditions, fluctuating demands of various stakeholders or maturing technologies. The ability to anticipate and manage prevailing volatility is supported by short-cycled respectively agile development.

After the relevant influencing factors were defined, the internal and external Agile Indicator can be derived in order to evaluate the identified product scopes. The quantitative evaluation of product scopes with Agile Indicators enables a definite and reproducible assessment of product scopes for an agile or plan-driven development. The derivation takes places in three steps.

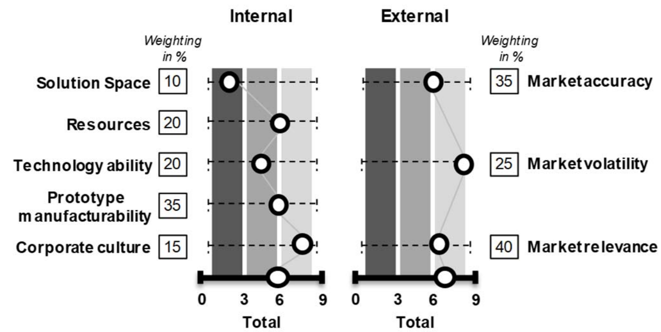


Fig. 4 Derivation of the internal and external Agile Indicator [10]

In a first step, the individual internal and external influencing factors must be weighted with regard to their relevance for the overall development project. An internal cross-domain team should conduct this weighting to ensure expertise in a wide range of business areas. The weighting requires profound knowledge of the company's own strengths and weaknesses as well as a systematic collection, analysis and use of relevant information on markets, competitors and economic conditions. Methodically, this weighting can be conducted by a paired comparison. In a second step, each individual product scope is evaluated on a ratio scale of 0, 3, 6 and 9. If a product scope is not suitable for agile development at all, it is assessed with 0. If it is fully suitable for agile development it is assessed with 9. The classification 3 and 6 constitute respective gradations. In order to ensure a comparability between the evaluation of individual product scopes the influencing factors have to be operationalized (see [11] [12]). Finally, the internal and external Agile Indicator can be aggregated by a summation of the weighted individual evaluation for each influencing factor. An exemplary application is illustrated in Fig. 4. The corresponding equation is shown below.

$$I_k = \sum_{i=1}^{n} W_i \times E_i \qquad (1)$$

I_k	Agile Indicator (k = index for internal/external)
i	Considered influencing factor
n	Number of influencing factors
W	Weighting of influencing factor
E	Evaluation of product scope

The internal Agile Indicator quantifies the internal preconditions and capabilities for an agile development of a product scope, while the external Agile Indicator reflects the market respectively customers' perspective. In combination, these two indicators allow a definite interpretation of the suitability evaluation of product scopes for an agile or plan-driven development. Therefore, the values of the internal and external Agile Indicator are marked in the "Agile Portfolio", which is presented in Fig. 5.

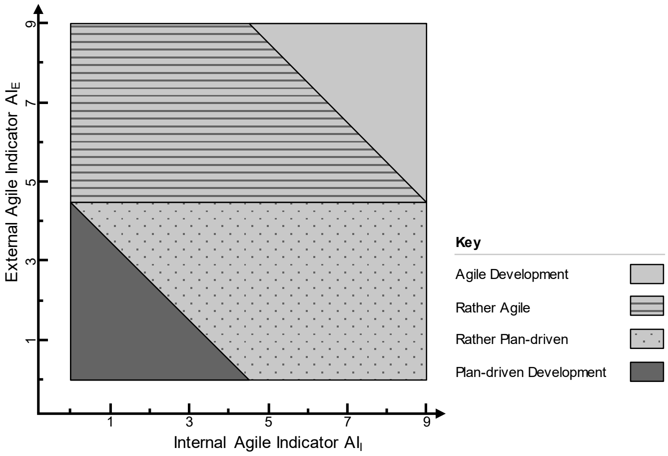


Fig. 5 Agile Portfolio

The Agile Portfolio constitutes the basis for the interpretation and analysis regarding the selection of agile product scopes in the following section C. The classification on the ordinate is based on the derived external Agile Indicator and the classification on the abscissa is based on the derived internal Agile Indicator. Depending on the numerical values, the portfolio defines four areas that can be interpreted as development recommendations. Product scopes, which are assessed with both a high internal and external Agile Indicator are in the light grey area "Agile Development" and should be developed by means of agile methods. Product scopes with a low internal and external evaluation, marked in the dark grey area "Plan-driven Development", should be developed according to plan-driven approaches like the Stage-Gate process. The resulting subdivision of the portfolio into areas with diagonal boundaries represents a well-established procedure in the scientific literature of portfolio analysis [25]. The remaining part of the portfolio is divided into an area "Rather Agile" with an external Agile Indicator above 4,5 and an area "Rather Plan-driven" with an external Agile Indicator below 4,5. The external Agile Indicator is given greater importance than the internal Agile Indicator, because it cannot be influenced by internal resources. Product scopes evaluated in these areas are typically developed according to the recommendation. However, a prioritization and an analysis of interdependencies between product scopes with different development recommendations is required for a final selection of agile product scopes. Therefore, the recommendation obtained from the Agile Indicators forms the basis for the next section of the methodology.

C. Selection of agile product scopes in hybrid development

After product scopes are evaluated individually by means of Agile Indicators and a development recommendation is given in the Agile Portfolio, this section aims at a conclusive selection of agile or plan-driven product scopes in a hybrid development. Therefore, interdependencies on the component level between individual product scopes are analyzed. Conflicts may arise, for instance, if a product scope A, with a recommendation for agile development, and a product scope B, with a recommendation for plan-driven development, use at least one common component. In order to design a hybrid development project consisting of plan-driven and agile product scopes as efficiently as possible, these interdependencies have to be resolved. This can essentially be achieved in three different ways. Firstly, the dependent product scopes are developed separately, i.e. in parallel. Thereby, the dependent components of the product structure are consciously developed "twice" and a later adaption development of the components is taken into account. In the second case, an originally plan-driven evaluated product scope is developed agilely (in context of the present paper, this is referred to as "agilization"). This is particularly advisable, if the uncertainty underlying the agile product scope is substantial. In the third case, a product scope with an evaluation for an agile approach is developed in a plan-driven way, contrary to its recommendation (in context of the present paper, this is referred to as "conceptualization"). In this case, the uncertainty of both product scopes is not particularly high, which makes a compulsive agile development of both product scopes or a separation obsolete. The following subsections describe how a decision can be made systematically for one of the three outlined cases of separation, agilization and conceptualization.

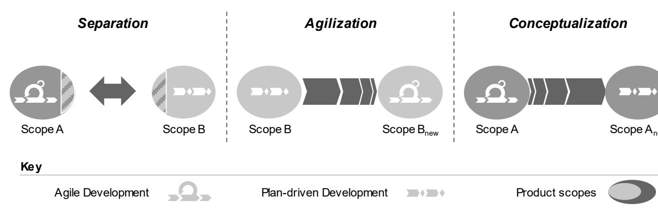


Fig. 6 Possibilities of resolving interdependencies

First of all, interdependencies between different product scopes on the component level have to be identified. As complex technical products are at best "nearly decomposable", they are never devoid of (potentially unknown) intermodular interdependencies that impede their fully individual development. In order to identify interdependencies between product scopes on a component level, the DSM (C/C) introduced in section A of the present methodology, can be used. In section A, the DSM has been used to derive concrete product scopes from classified development question with help of cluster algorithms. An interdependency occurs, if two clusters have an intersection in form of one or more components that are part of both clusters. In order to be able to decide whether interdependent product scopes are considered in a coherent scope or whether the components should be developed in parallel, further analysis of the intersection is necessary.

In a next step, the intersection is specified through a description of its complexity. The complexity of an intersection results from the combination of two variables: number of components and number of product scopes. The number of components specifies how many components are included in an intersection (value range [1;n]). The number of product scopes describes how many product scopes utilize the components in an intersection (value range [2;m]).

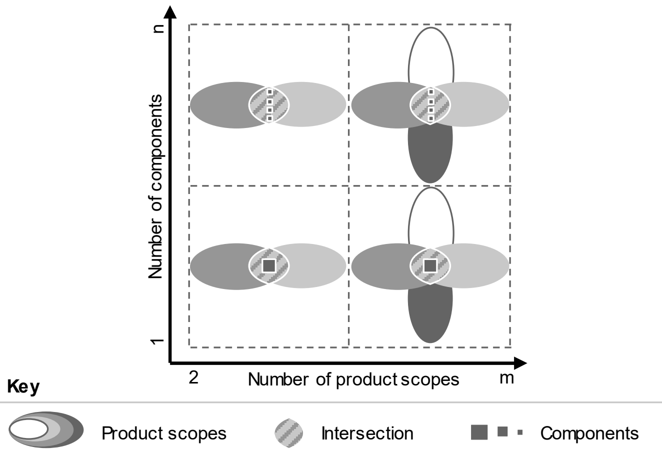


Fig. 7 Complexity of an intersection of interdependent product scopes

Two essential cases of the complexity of an intersection can be deduced. There are intersections with a small number of dependent components and product scopes, which have a low level of complexity (case 1). For instance, if two product scopes are dependent on a single component, the resulting complexity is low. However, if the number of the two variables “number of components” and “number of product scopes” increases, the complexity of the intersection increases as well (case 2). The determination of whether an intersection can be assigned to case 1 or case 2 has to be made on the basis of expert assessments in the individual case and cannot be quantified objectively.

Based on the identification and complexity description of interdependent product scopes, a definite recommendation for agile or plan-driven development has to be made. Therefore, an examination of the underlying development questions of interdependent product scopes can lead to an ad-hoc decision for the resolution of the intersection. Consequently, the types of development questions classified in section A of the present paper are considered. As conceptual and design development questions focus on the shape, proportions and surfaces of a product, functional and technical development questions focus on the functionality and the verification of the overall product [26]. This results in different maturity requirements for the components to be developed [27]. Consequently, interdependencies between these two maturity types of conceptual and geometrical development product scopes on the one hand, and functional and technical product scopes on the other hand do not have to be considered any further, but can be developed separately.

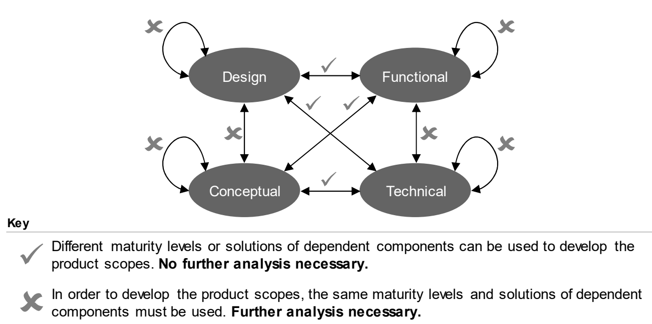
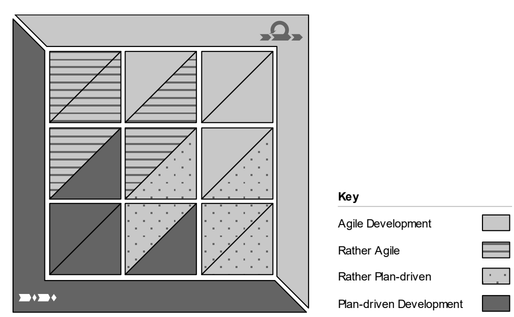


Fig. 8 Examination of types of development questions

Product scopes that have been identified for further analysis must be examined with regard to their internal and external Agile Indicators in order to resolve the prevailing intersection conflict and thus unambiguously classify them for agile or plan-driven development. For this purpose, the Agile Portfolio presented in section B of this methodology is used. According to GAUSEMEIER, portfolio analyses are a widespread and widely used method for the derivation of recommendations for action [28]. The consideration of a 1:1 interdependency between two product scopes, which can each be assigned to one of the four sectors of the Agile Portfolio, results in ten different possible combinations (see Fig. 9). In this context, it is crucial to examine, which recommendation (e. g. same or different) and with which characteristics (e. g. strong or weak) the individual combination are based on. Interdependent *product scopes with the same development recommendation* can be **developed according to the recommendation**, since the development recommendations of the individual product scopes do not have a negative influence on each other. This is the case, for example, if two product scopes with a clear agile development recommendation interact.

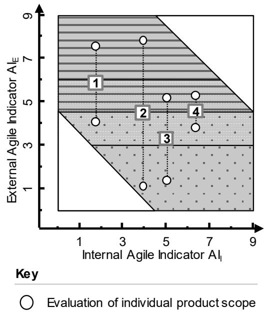
Moreover, in the case of *product scopes with different development recommendation but a strong characteristic* for an agile and plan-driven realization, there is no need for further analysis either. **The product scopes should be developed separately** according to their recommendation, since a joint development would jeopardize a reduction of uncertainty, which initially was intended by an agile evaluation of one of these product scopes. However, interdependent product scopes with different development recommendations and a weak characteristic in terms of a clear evaluation for agile or plan-driven realization have to be analyzed in-depth in the individual case (see table in Fig. 9).



Combination	Recommendation	Characteristic	Description
	different	strong - strong	**No further analysis necessary!** Separate development.
	different	weak - weak	**In-depth analysis necessary!** Which development recommendation should be implemented?
	different	strong - weak	**In-depth analysis necessary!** Which development recommendation should be implemented?

Fig. 9 Visualization of possible combinations of interdependent product scopes in the Agile Portfolio

For the consideration of combinations of *product scopes with different recommendations and a weak characteristic*, the areas of the Agile Portfolio should be analyzed with regard to the evaluation of the external Agile Indicators. As described in section B of the present methodology, the transition from a rather plan-driven recommendation to a rather agile recommendation is defined for an external Agile Indicator with a value of 4.5. Due to the fact that a better founded recommendation can be made for values that are further away from this mean value on the axis of the external indicator, the portfolio should be subdivided directly above and below the mean value (each ± 1.5). This results in **four possible combination scenarios for interdependent product scopes with different recommendations** and a weak characteristic (see Fig. 10.).



Case	Value Range AI_E	Recommendation
1	$AI_{E1} > 6 \wedge 4{,}5 > AI_{E2} > 3$	*Agilization*
2	$AI_{E1} > 6 \wedge AI_{E2} < 3$	*Separation*
3	$6 > AI_{E1} > 4{,}5 \wedge AI_{E2} < 3$	*Conceptualization*
4	$6 > AI_{E1} > 4{,}5 \wedge 4{,}5 > AI_{E2} > 3$	*Individual Examination*

Fig. 10 Resolution of intersections with different development recommendation and weak characteristic

In the first case, in which an agile development of the interdependent product scopes is aspired, a combination of a product scope with an external Agile Indicator in the value range [AI_{E1}>6] and another product scope with an external Agile Indicator in the value range [4,5<AI_{E2}<3] is present. **These product scopes should preferably be developed in an agile way**. In the second case, the external Agile Indicators of the interdependent product scopes range between the values [AI_{E1}>6] on the one hand and [AI_{E2}<3] on the other hand, and thus exhibit a significant difference, which is why **a separate development should be targeted**. The third case is characterized by a value range [6>AI_{E1}>4,5] of one product scope and thus indicates a rather agile development and a second value range [AI_{E2}<3] of another product scope that indicates a definite plan-driven development. In this case, **a conceptualization should be applied** in terms of a plan-driven development for both product scopes. Finally, the fourth case does not allow a clear recommendation due to the value ranges [6>AI_{E1}>4,5] and [4,5>AI_{E2}>3] of the product scopes. **In this case, neither a definite recommendation for agile development nor for plan-driven development can be made based on the evaluation with Agile Indicators in** the Agile Portfolio. Therefore, a decision has to be made with help of an interdisciplinary expert team in each individual case.

Finally, after the combination of product scopes with different development recommendations and a weak characteristic has been considered, possible intersections between *product scopes with different development recommendations and a "strong - weak" characteristic* have to be resolved (see Fig. 9). In order to be able to resolve the intersection between product scopes with different recommendations and a strong - weak characteristic, it is not sufficient to consider the combination of areas of the agile portfolio singularly. Therefore, the complexity of an intersection, which was derived previously, must be taken into account. The consideration of the complexity of an intersection helps answering the question of whether a separation or combination into a single agile or plan-driven product scope is more reasonable for the overall development project. In particular, it is necessary to weigh up the additional expenditure for coordinating and harmonizing individual product scopes resulting from a possible separation against the benefits of an agile realization and the associated early elimination of uncertainty. Therefore, the two previously defined cases of complexity of an intersection are taken into consideration. The first case (low level of complexity) is characterized by **a small number of interdependent components and product scopes,** which offers an appropriate balance between coordination effort and development benefits and **should be separated.** In the second case, the two variables "number of components" as well as "number of product scopes" rise and the coordination effort within a possibly separated development increases. Therefore, **intersections with a high level of complexity should be combined in a product scope and subsequently implemented either agile or plan-driven**. The decision on whether the product scopes, which are characterized by a high degree of complexity within the intersection should be implemented either plan-driven or agile, can be determined by considering the strong recommendation of the singular evaluation of the product scopes. If there is a strong recommendation for a plan-driven development of one of the interdependent product scopes, the combined product scope should be developed according to the plan-driven recommendation. This is based on the fact, that product scopes resulting from the combination show a clear tendency towards the area of a plan-driven development in the Agile Portfolio. The same applies to the case of a strong agile recommendation of one of the interdependent product scopes. In this case, a strong tendency towards the area of agile development can be determined, which is why the combined product scope should be realized in an agile way.

Intersections between individual product scopes with different development recommendation have been illustrated by means of identification and description of interdependencies and could be resolved by a consecutive detailed analysis. As a result, there are product scopes with a definite recommendation for either agile or plan-driven development, which can be clearly marked in the Agile Portfolio. The methodology presented in this paper enables determining the application suitability of agile development methods for technical products and ensures the right degree of agility in a development project.

IV. Conclusion

In order to apply agile methods in a hybrid development project for technical products, manufacturing companies have to ensure the right degree of agility. Therefore, the application suitability of agile development methods for individual product scopes is examined. Consequently, prevailing uncertainties in a development project can be successively reduced. The present methodology enables the deduction of classified development questions resulting from inadequately described requirements and the systematic derivation of product scopes, which are based on existing development questions. The evaluation of individual product scopes regarding their suitability for agile development is performed by the application of Agile Indicators, which enable a market and company-specific, reproducible assessment. Furthermore, a selection of agile and plan-driven product scopes can be made based on the analysis and resolution of interdependencies between product scopes.

The presented methodology is intended to be applied to mechatronic products in the product development phase. By focusing on the product development phase, it is ensured that requirements are not yet known or incomplete, which makes it necessary to consider system dependencies of requirements, development questions and physical components. In addition, the methodology can be applied across all industries in order to ensure application in the widest possible range.

The validation and further investigation of the presented methodology is currently part of the research activities of the authors. The transfer of agile methods towards the development of technical products is a central topic of investigation within the research project "Design of agile development processes for technical systems" funded by the Deutsche Forschungsgemeinschaft (DFG).

ACKNOWLEDGEMENT

The presented results have been developed within the research project "Design of agile development processes for technical systems" funded by the Deutsche Forschungsgemeinschaft (DFG).

REFERENCES

[1] R. G. Cooper, "Idea-to-Launch Gating Systems. Better, Faster, and More Agile", In: Research-Technology Management, vol. 60., no. 1, pp. 48–52, 2017.

[2] C. Hoffmann, S. Lennerts, C. Schmitz, W. Stölzle, "Business Innovation: Das St. Galler Modell", p. 283, 2016.

[3] R. G. Cooper, "Stage-Gate systems: A new tool for managing new products", Business Horizons, vol. 33., no. 3, pp. 44–54, 1990.

[4] A. F. Sommer, "Improved Product Development Performance through Agile/Stage-Gate Hybrids", p.1, 2015.

[5] P. Serrador, "Does Agile work? - A quantitive analysis of agile project success", p.1, 2015.

[6] R. G. Cooper, A. F. Sommer, "The Agile–Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity", Journal of Product Innovation Management, 2016.

[7] A. Komus, M. Kuberg, "Studie Status Quo Agile 2016/2017", In: GPM - Deutsche Gesellschaft für Projektmanagement e.V., 2017.

[8] R. G. Cooper, T. Vedsmand, S. Kielgast, "Integrating Agile with Stage-Gate®. How New Agile-Scrum Methods Lead to Faster and Better Innovation", In: Innovation Management, 2016.

[9] G. Schuh, M. Riesener, F. Diels, "Determining the Application Suitability of Agile Methods for Highly Interdependent Products", Management of Engineering and Technology (PICMET), pp. 1-10, 2017.

[10] G. Schuh, M. Riesener, F. Diels, "Structuring highly iterative product development projects by Using HIP-indicators", Industrial Engineering and Engineering Management (IEEM), pp. 1171-1175, 2016.

[11] G. Schuh, S. Rudolf, F. Diels, "Methodology for the evaluation and selection of the suitability of highly iterative product devel-opment methods for individual segments of an overall develop-ment project", Industrial Engineering and Engineering Management (IEEM), pp. 661-665, 2015.

[12] F. Diels, S. Rudolf, G. Schuh, "Highly iterative product development process for engineering projects", Applied Mechanics & Materials, pp. 532-539, 2015.

[13] E. C. Conforto, D. C. Amaral, "Agile project management and stage-gate model - A hybrid framework for technology-based companies", Journal of Engineering and Technology Management, 2016.

[14] J. Feldhusen,.M. Löwer, F. Bungert, "Agile Methods for Design to Customer", pp. 451-460, 2009.

[15] T. P. Klein, G. Reinhart, "Towards Agile Engineering of Mechatronic Systems in Machinery and Plant Construction",Procedia CIRP, vol. 52, pp. 68-73, 2016.

[16] U. Eklund, C. Berger, "Scaling Agile Development in Mechatronic Organizations - A Comparative Case Study". In: arXiv preprint arXiv:1703.00206, 2017.

[17] J. Feldhusen et al., "Konstruktionslehre", Pahl/Beitz, p. 137, 2013.

[18] P. Link, "Agile Methoden im Produkt-Lifecycle-Prozess", p. 72, 2014.

[19] E. Moeller, "Handbuch Konstruktionswerkstoffe - Auswahl, Eigenschaften, Anwendung". München: Hanser, 2013.

[20] G. Schuh, M. Riesener, S. Schloesser, "Prototyping in Highly-Iterative Product Development for Technical Systems", Advanced Materials Research, 2017, unpublished.

[21] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems", 292ff., 2001.

[22] T. U. Pimmler et al., "Integration Analysis of Product Decompositions", p. 347. 1994.

[23] S. S. Altus et al., "A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems", 1996.

[24] A. Picot et al., Modularität in der verteilten Entwicklung komplexer Systeme, p. 223, 2007.

[25] Pfeiffer et al., "Technologie-Portfolio zum Management strategischer Zukunftsgeschäftsfelder", 1987.

[26] F. Klocke, "Generative Fertigungsverfahren", In: Fertigungsverfahren 5, Springer Vieweg, Berlin, pp. 127-170, 2015.

[27] A. Gebhardt, "Additive Fertigungsverfahren: Additive Manufacturing und 3D-Drucken für Prototyping-Tooling-Produktion", Hanser Verlag, p. 364, 2016.

[28] J. Gausemeier, A. Fink, O. Schlake, "Scenario management: An approach to develop future potentials", Technological Forecasting and Social Change, vol. 59, no. 2, pp. 111-130, 1998.