designs

MDPI

Article

Agile Powertrain Development: Considerations to Incorporate Agile Principles

Andreas Lukas, Oliver Moerth-Teo *, Lukas Schwarz, Hans P. Schnöll, Matthias Wolf and Christian Ramsauer *

Institute of Innovation and Industrial Management, Graz University of Technology, 8010 Graz, Austria; andreas.lukas4@gmail.com (A.L.); lschwarz1989@gmail.com (L.S.); schnoell@tugraz.at (H.P.S.); matthias.wolf@tugraz.at (M.W.)

* Correspondence: oliver.moerth@tugraz.at (O.M.-T.); christian.ramsauer@tugraz.at (C.R.); Tel.: +43-316-873-7091 (O.M.-T.)

Abstract: Pressure to increase effectiveness and efficiency drives the product development process. The software industry has been using agile development approaches due to their advantages when coping with uncertainties and simultaneously increasing value. Therefore, the following work deals with the considerations of agile powertrain development. In order to identify important conditions for the incorporation of agile principles into powertrain development, semistructured interviews were conducted with experts from an engineering company. Having identified agile development challenges, advantages, and principles, the next step is comparing agile principles with the current principles of conventional powertrain development. Furthermore, a basic understanding of the powertrain development process regarding the incorporation of agile principles is established, the main attributes of powertrain development are investigated, and the key disciplines involved are identified. Finally, a model is created to assess whether the application of an agile or conventional development approach is more suitable.

Keywords: product development process; agile powertrain development; incorporation of agile principles

Citation: Lukas, A.; Moerth-Teo, O.; Schwarz, L.; Schnöll, H.P.; Wolf, M.; Ramsauer, C. Agile Powertrain Development: Considerations to Incorporate Agile Principles. *Designs* **2021**, *5*, 60. https://doi.org/10.3390/designs5040060

Academic Editors: Roberto Gabbrielli, Marcello Braglia, Leonardo Marrazzini, Marco Frosolini and Luca Padellini

Received: 18 August 2021
Accepted: 23 September 2021
Published: 27 September 2021

1. Introduction

The world as we know it is affected by political, economic, social, technological, and environmental trends. The United States War College first used the acronym "VUCA" in 1987 to describe the volatility, uncertainty, complexity, and ambiguity associated with these trends. Staying competitive in such an environment requires appropriate anticipation, understanding, preparation of strategies, and implementation of effective management interventions [1]. The effects caused by VUCA are a faster-paced, more competitive, and global, but less predictable world [2], or, in other words, a more chaotic world [3]. As a result, industrial companies are facing numerous challenges such as shortened product life cycles, increased product variants, faster-changing customer requirements, and more [4]. In particular, powertrain systems will not be spared from the uncertainties caused by new technologies for propulsion and storage systems. Schuh et al. (2017) stated that "classic" so-called plan-driven development processes, which are still primarily used in powertrain development, are too rigid to react sufficiently to the new dynamic environments as they are unable to adapt to changes in the market. Thus, there is a need to make the development process more flexible to respond to these dynamics. One possibility is the incorporation of agile development principles from the software industry. Through short cyclical development loops, results are continuously evaluated and sudden changes can be detected more easily and earlier to avoid costly corrections. Unfortunately, the development of mechatronic systems, which includes powertrain systems, leads to restrictions on the adoption of such approaches: long tool lead times, extensive efforts to plan and implement tests, high organizational complexity with given hierarchical levels, the breaking of human habits, etc. [5]. Nevertheless, Žužek et al. (2020) claimed that several companies from different industries outside the software sector have already adopted agile principles for their product development approaches despite those restrictions. The authors demonstrated that only implementing separate principles at an SME allows for achieving significant benefits: improved communication, faster detection of discrepancies, more effective problem-solving, and greater flexibility [6]. Volkswagen, for example, is certainly committed to agile working methods as more people in the company are using related principles. In this respect, Volkswagen emphasizes the core component as being "independently organized teams" that handle project phases, while managers concentrate on defining goals and prioritizing requirements [7]. Hence, more and more attention is being paid to agile development approaches in the automotive sector. However, it is not yet clear whether agile approaches can bring about benefits in the development of powertrain systems and what considerations are required to make them a success.

Based on this problem statement, the relevant research questions were formulated as follows:

- Which powertrain development considerations are important for the incorporation of agile principles?
- How can the incorporation of agile development into an existing powertrain development process be supported?

In order to support the practical application of agile principles in the powertrain development process, a close cooperation with an industry partner in the relevant business sector was built on the basis of a single case study. Besides an extensive literature review, semistructured interviews were conducted with relevant engineers and decision makers for data collection in a qualitative manner. As an agile approach has already been implemented in one division of this company (the subsequently named division Alpha), the challenges, advantages, and principles of agile product development have been identified. This is supplemented by the agile considerations, attributes, and disciplines in the powertrain development division (subsequently named division Beta). The gathered data are structured and we analyze the results to come to conclusions on the considerations for agile powertrain development. Subsequently, the development principles in the two divisions Alpha and Beta are compared. In the final step, a procedural model is formulated as support for the incorporation of agile principles in powertrain development.

2. Related Work

2.1. Conventional Product Development Approaches in the Automotive Industry

It is important to mention that the term "conventional" in this paper refers to all product development approaches that are not seen as "agile" in the literature. According to Lührig (2006), the activity of product development can be described as follows: "Product development is the targeted transformation of an idea into a combination of goods or services for commercial purposes, carried out by one or more persons (organization). This requires that the idea to be transformed is technically feasible and that there are sufficient marketing opportunities for the product to be developed" [8]. First, there is always a predevelopment or product planning phase that serves as the starting point. Within this period, ideas (for products, process, or innovation) are compared with the corporate strategy and other important strategies—for instance, regarding the product range. Furthermore, market analyses are carried out and general conditions such as time-to-market or quantities are determined. Subsequently, the project enters the definition phase, which can be divided into the target and concept definition. Here, content from strategic product planning is translated into requirements for the entire vehicle. The target definition includes the setting of deadlines and development milestones. On the other hand, the concept definition concludes with the so-called specification sheet, which contains all requirements regarding the functions, quality, features, and general conditions of the vehicle [8,9]. In the product development phase, both concept development and series development take place. First, the requirements from the specification sheet are translated into technical solutions, ranging from the vehicle level down to the component level. Solutions that sufficiently fulfil their related requirements are evaluated against the technical and economic criteria before their actual implementation. In the concept phase, the first drawings or CAD designs are created, which are then transferred to the series development. When an overall vehicle concept is defined at the end of the concept development phase, a so-called "design freeze" occurs and the systematic integration of the components can be transferred to the vehicle. Series development characterizes the transition phase to production. The detailed process and product description are at the end of the series development, after which the integration of the components is further coordinated. While in the preseries phase only series production conditions are tested, the SOP (Start of Production) launches the series production [8]. A general illustration of these phases can be seen in Figure 1.

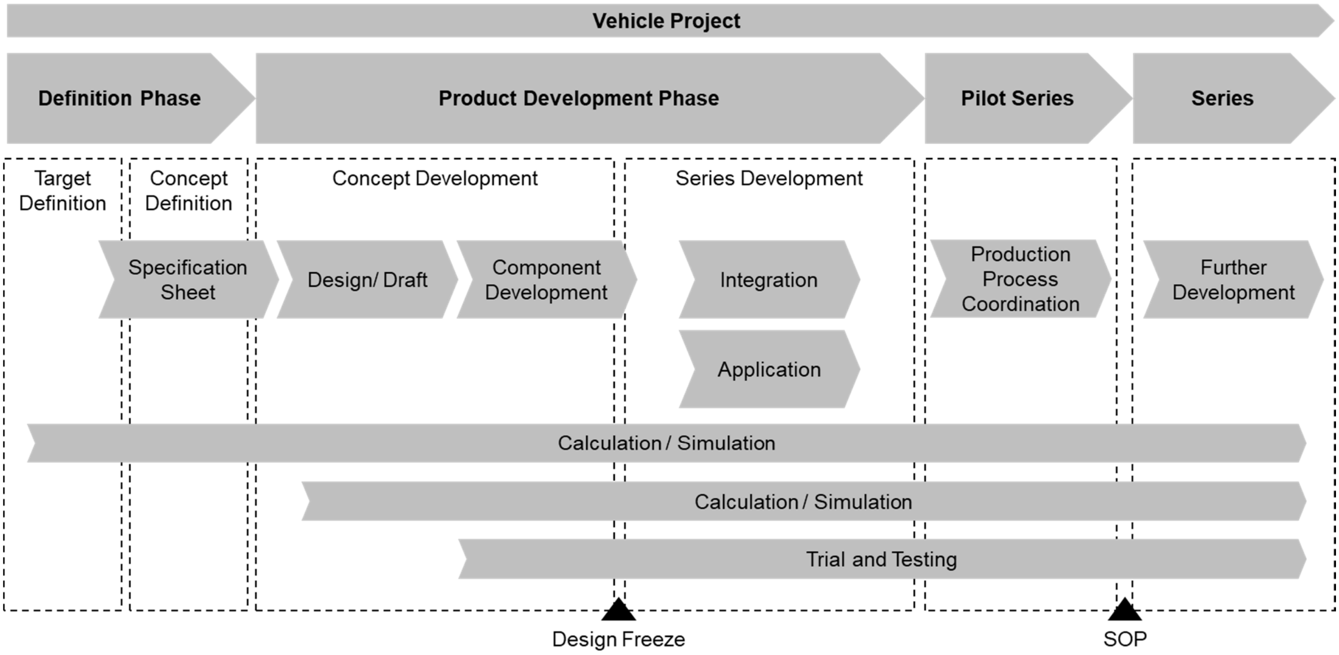

Figure 1. General product development process in the automotive industry (adapted from [8,9]).

2.1.1. VDI 2221

VDI 2221, also known as methodology for the development and design of technical systems and products, provides targeted methodical instructions and defines work steps and desired results. It refers especially to the discipline of mechanical engineering and provides an overview of methods that can be applied in various development projects. The methodology is divided into seven main steps (depending on the development task, these steps can be run several times iteratively), from which seven results are generated. The steps and related results are described as follows [10]:

- Step 1—Clarification and definition of the problem: Step one is the complete clarification of the customer's requirements or internal requirements. This also includes defining the problem from the constructor's perspective. The result of the first step is the requirements list.
- Step 2—The determination of functions and their structure: First, the main function that the product needs to fulfil is determined; then, subfunctions are defined. By structuring and combining these functions, one obtains the result of the work step as a function structure.
- Step 3—Search for solution principles and their structure: In step three, solutions are sought for all defined functions and implemented through effects. The result of this step is the principal solution, such as a schematic layout.

- Step 4—Division into realizable modules: Fourth, modules are created to make the principal solution feasible. Individual solution systems and subsystems as well as interfaces between the modules are already defined.
- Step 5—Design of the most important modules: Here, a concrete dimensioning of the most important modules takes place as far as the state of knowledge allows. The results of this step are preliminary designs, general layouts, or preliminary drafts such as rough-scale drawings.
- Step 6—Design of the entire product: In this step, final shapes or detailed designs are created by further refining the preliminary designs with details and adding the modules that have not yet been completed. The result is the overall design with scale drawings, parts lists, and so on.
- Step 7—Compilation of design and utility data: This is where decisions are made about the manufacturing and application of the product, as well as the usage specifications of the product. The result is the consolidation of all data, such as CAD drawings for product documentation.

2.1.2. Cooper's Stage Gate

Cooper's Stage-Gate process has been the basis of several project management and product development models since the 1980s in all kinds of business and industry sectors [11]. Generally, those Stage-Gate models are seen as plan-based approaches, as Cooper himself states that proper planning beforehand can help a company to develop a new product more efficiently. The process itself consists of 4–7 stages and gates. With each stage, the available information becomes clearer and the risk can be minimized, whereas each step is more expensive than the previous one. Before a stage can be entered, going through a so-called gate, which can be seen as a quality control checkpoint, is necessary. A gate is always characterized by distinctive deliverables that allow for going in, and certain criteria that need to be fulfilled for going out of it. Therefore, decisions like "Go," "Kill," "Hold," or "Recycle" need to be done by a so-called gatekeeper. Cooper states that the more work that is done in advance (or upfront), the fewer changes will be seen as necessary in hindsight [12]. The advantages of stage-gate processes are faster development, fewer errors, better decisions, and a higher focus on the essential tasks.

2.1.3. V Model

The VDI 2206 defines a standardized guideline for the development of mechatronic systems. Those mechatronic systems require interdisciplinary collaboration and cross-domain communication of the disciplines of mechanical engineering, electrical engineering, and information technology. The guideline's intention is to serve as a supplement for the VDI 2221, which is more focused on methodological design and development. The main objective of VDI 2206 is to provide procedures, methods, and tools for the early development phase, with a focus on system design, to provide an assured concept in the end. The resulting solutions are always verified and validated. However, the validation can be carried out with varying degrees of accuracy based on virtual or physical prototypes. VDI 2206 is assumed as a flexible procedural model that is based on three main elements [13]:

- Micro-level: general problem-solving cycle;
- Macro-level: V-Model;
- Predefined process steps for conduction of recurrent development steps.

2.2. Agile Product Development Approach

From a historical perspective, product development has always been dominated by the mechanical domain, whereas additional domains such as electrical engineering were added later [13]. Today, companies are confronted with a lot more external pressure and internal complexity, which results in the need to act fast and be more flexible in product development. Due to the pace of change, conventional approaches where business departments or R&D departments develop products only on the basis of defined specification sheets are no longer feasible for generic product development [11]. Shrinking product lifecycles—while the Golf I was built and sold for nine years in Germany, this shrank to five years for the Golf V—more intense competition, saturated markets, and increasing globalization force companies to be more innovative in order to stay competitive. In particular, big companies see themselves as more and more in danger due to radical innovation [5]. As agile methods promise rapid value and responsiveness to change [14], agility in product development is seen as beneficial and essential [11]. Combining agile and stage-gate principles is often referred to as a hybrid process. Results from already successful implementations at high technology-based companies indicate positive impacts on the project and product performance through balancing stability with flexibility [15].

The term "agility" is not unambiguously defined in the literature. Conforto et al. (2016) emphasized the problem of a missing common definition of the term "agility" in project management such that different understandings result in various interpretations. Having conducted a study examining the related literature, they defined agility in project management as "the project team's ability to quickly change the project plan as a response to customer or stakeholders needs, market or technology demands in order to achieve better project and product performance in an innovative and dynamic project environment" [16]. Ramsauer et al. (2017) define agility for manufacturing companies in a similar way, as "the capability of a production company to proactively prepare for uncertainties and to react quickly to changes in order to optimize the economic situation of a company, by using the entire value chain" [4]. A further definition from Schuh (2019) of agile product development is "a procedure in which development scopes are worked on by self-organized and interdisciplinary teams in iterations, while incorporating agile principles. The overriding goal is to improve the fulfilment of customer needs while reducing the development time simultaneously" [17]. For the purposes of this paper, agility in product development refers to agility as a capability of either a company [4] or a project team [15,16].

In the context of product development, agile methods were first used in software projects [11]. Kantelberg (2018) notes that shorter development times resulted in a need to also shorten software development. At the same time, the complexity of software projects increased drastically, making it difficult to use conventional methods. With this background, agile development methods emerged in the software industry [18]. Before that, Takeuchi and Nonaka (1986) had noted the importance of agile characteristics such as built-in instability, self-organized teams, overlapping development phases, multilearning, subtle control, or organizational transfer of learning, and used the term "Scrum" for them [19]. Cooper and Sommer (2016) state that there are at least 26 different agile approaches, and Scrum is the most popular framework for product development [11]. Klein (2016) summarizes the key approaches as follows [20]:

- Dynamic Systems Development: framework for the application of a prototypical procedural model for the rapid development of applications under consideration of defined principles.
- Scrum: Widely used framework in the software industry for project management and system development.
- Feature-Driven Development: A given process sequence with five phases and best practices iterative software development.
- Crystal: An application specific selection and adaption of a process model based on criteria of project criticality and team size.
- Extreme Programming: A synthesis of ideas and approaches from existing methods for development and planning activities for software.
- Adaptive Software Development: Adaptive life cycle for software development including development philosophy for change management.
- Agile Modeling: Values, principles, and methods for efficient modeling in combination with agile process models.
- Lean Software Development: Describes a transfer of the lean philosophy from the Toyota production system and IT into seven principles for software development.

- Agile Unified Process: Hybrid modeling approach of the Rational Unified Process with agile software development.
- Usability Driven Development: Iterative driven development process with a focus on the usability of the system.
- Kanban: Software development in consideration of throughput times, bottlenecks, etc., comparable to the method of the same name for lean production.

In practice, a clear differentiation between plan-driven and agile project management approaches can hardly be found. Cobb (2011) noted that there are many good reasons to select a plan-driven method for certain projects, depending on the risk and complexity. However, there are ways to incorporate agile principles into plan-driven approaches to make them more agile, and the possibility of applying plan-driven principles to agile approaches to achieve a higher level of predictability and control [21]. Boehm and Turner (2004) differentiate between agile and plan-driven principles in software projects. They show the "home grounds" of those approaches and define the criteria for success in a business environment. Additionally, they refer to an increasing risk of failure the more a particular project's conditions differ from the home-ground conditions of the applied approach [14]. An extensive literature review was performed to define the characteristics for conventional and agile product development. A list of the most relevant characteristics is provided in Table 1, where criteria 1–4 are related to the project, 5–10 are related to the product, 11–13 are related to the environment, 14–15 are related to the customer, and 16–18 are related to the team.

Table 1. Characteristics of conventional and agile development approaches.

No.	Criteria	Conventional	Agile	Source
1	Project goal	Predictability; stability; high assurance; clear Management of complicatedness	Rapid value; responding to change; not clear Management of complexity	[14,22–24]
2	Solution	Clear	Not clear	[22–25]
3	Goal on design	Optimization	Adaption; flexibility and responsiveness	[26,27]
4	Project (and product) size and duration	Large products; bigger projects (years)	Small products; smaller projects (months)	[3,14,20,25,28–31]
5	Product requirements	Foreseeable evolution requirements; stable requirements and specifications; clear initial requirements Priority and number of requirements stays the same	Unforeseen change; change often during project; unclear Priority and number of requirements is volatile	[3,11,14,20,24,28–31]
6	Product Definition	Established in detail upfront (>90%)	Partly established upfront (40–70%)	[2,29]
7	Product Type	Low product newness; new item in a product line; modification or improvement; renovation	High product newness; innovation; higher risk-initiative	[2,3,31]
8	Technology	Well known technology; mature; clear; in house	Some technical risks; newer technology but largely existing; may be new to company	[2,29]
9	Testing	Documented test plans and procedures	Executable test cases define requirements	[14]

Table 1. *Cont.*

No.	Criteria	Conventional	Agile	Source
10	Criticality	Extreme; Highly safety critical products; system failure consequences serious	Low; Non safety critical products; less critical systems	[3,11,14,25–31]
11	Environment	Stable; low change; predictable	Turbulent, high change; difficult to predict	[3,14,24–27,29–31]
12	Market and Competition + Market Size	Mature and well-known market; few market uncertainties or risks; large and defined; red ocean; many capable competitors	Existing and rapidly growing market; large potential to grow; many market uncertainties and risks; blue ocean; some early competitors	[2,29]
13	Collaboration and Communication	Low: business involvement only at start and end of project; formal communication	Continuous face-to-face business involvement; informal communication	[3,21,26,29]
14	Customer relation and availability	As-needed customer interactions; customer involvement at the beginning and end	Dedicated on site-customer interactions; customer involvement through whole process	[3,14,20,24,27,29,31,32]
15	Customer Needs	Well known and stable over time	Some known, some unarticulated Many unsolved customer problems and unresolved needs	[2,20,24]
16	Team size	Small teams	Large teams	[14,20,25,28–30]
17	Team Experience	Junior level; specialized; minimum communication skills	Senior (experienced); more experienced; good communication skills; interdisciplinary;	[3,11,14,25,29–31]
18	Team Members Dedication and Continuity	Team members on multiple projects concurrently; not accentuated; fluctuation expected; distributed teams	Team members dedicated to project; collocated and smaller teams	[2,3,28,30–32]

3. Considerations for Agile Powertrain Development

3.1. Research Approach

In order to support methodological fit, the relationship between the approach and the maturity of existing knowledge must be taken into account [33]. Arguing that the knowledge on agile powertrain development is rather nascent and intermediate, Ahlström (2016) states that the appropriate research approaches are case study, longitudinal field study, actions research, and clinical research [34]. Comparing the main characteristics of these approaches, a case study is preferable as the problem identified by the researcher is to be explored in reality [35]. In the next step, a single case design was chosen due to the uniqueness of the case [36] as well as the opportunity for a greater depth of observation [35]. The single case is represented by one engineering company in the automotive industry. This company is organized into several divisions, which differ not only in their business focus but also in their product development approach. For this study, two divisions, Alpha and Beta, were investigated. While Alpha deals with the development of electronic products for mass production based on an agile approach, Beta focuses on the development of powertrain systems through conventional methods. Investigating these two departments enabled us to compare agile and conventional development principles in a practical environment. This serves as a basis to support the incorporation of agile principles in the hardware-driven conventional powertrain development. Therefore, the challenges, advantages, and principles of agile product development were identified in

Alpha. This was supplemented by the agile considerations, attributes, and disciplines of powertrain development in Beta. An overview of this research approach is illustrated in Figure 2.

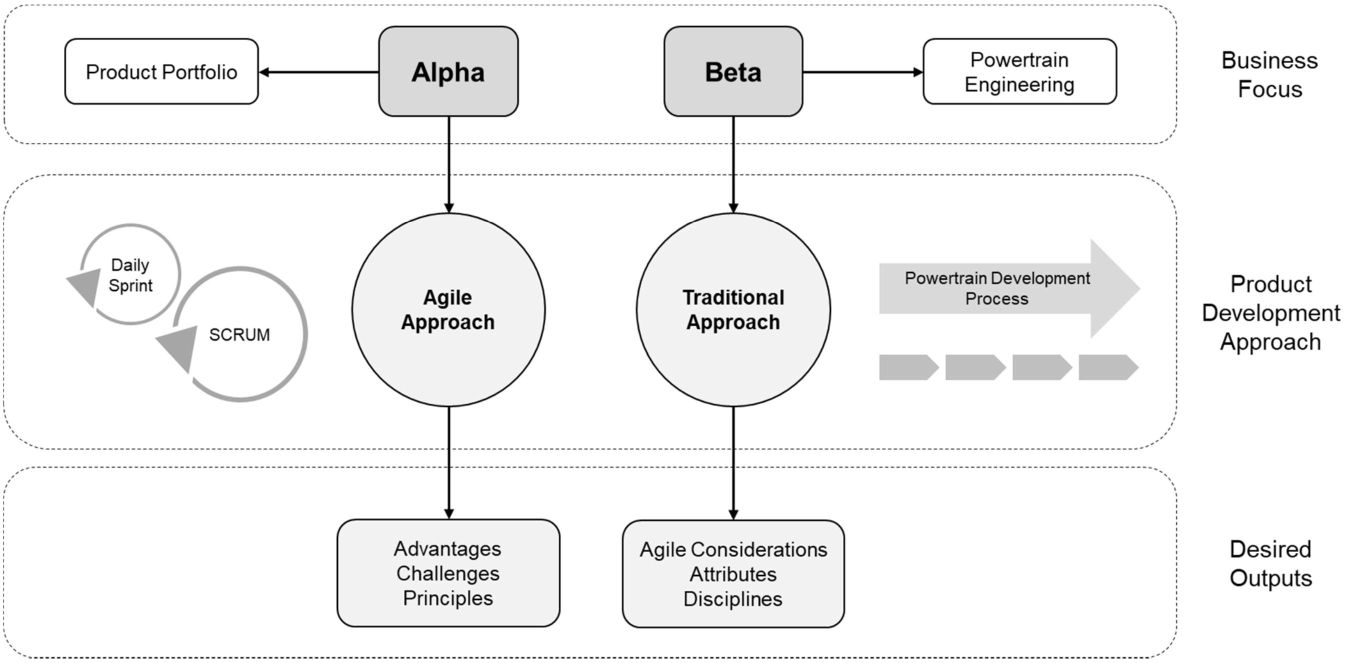
Product Portfolio
Alpha
Beta
Powertrain Engineering
Business Focus
Daily Sprint
SCRUM
Agile Approach
Traditional Approach
Powertrain Development Process
Product Development Approach
Advantages Challenges Principles
Agile Considerations Attributes Disciplines
Desired Outputs

Figure 2. Overview of research approach.

3.2. Alpha Division: Fundamentals of Agile Product Development

Alpha's business focus is not linked to powertrain development. Referring to the time when the interviews were conducted, an agile development approach based on the concept of Scrum was introduced in specific sections three years ago. However, a rather conventional process that defines certain phases, stages, and their deliverables is still the foundation of any product's development. As the basis of the agile development approach, this process defines—in the background—what needs to be done and documented at what time. In order to identify the advantages, challenges, and principles of agile product development, interviews with two project managers, one team leader for purchasing, one agile coach, and two scrum master of division Alpha were conducted, each with an approximate duration of one hour. While Table 2 presents a summary of the identified challenges, advantages, and principles, a more detailed description of them can be found in the following paragraphs.

3.2.1. Challenges

(1) As humans generally tend to reject change, considering the "human factor" is important for the successful incorporation of agile development principles. Due to the small number of team members, it is essential that all team members show full dedication. (2) Since the agile development method requires many meetings (e.g., daily stand-up, sprint review, sprint review, etc.), there is a lack of understanding of their needs, especially when the involvement of everyone is not always required. (3) When using the agile development method, it is essential to enforce strong contact and coordination of the customers and the suppliers. However, the practical realization of the information exchange is more difficult. (4) It is necessary to achieve a change in the mindset of the whole organization as this is one of the biggest enablers of the implementation of agile principles. (5) The identification of the required disciplines for a development project is essential for a proper team constellation. While coordination meetings are necessary for integration purposes, short intervals might be perceived as waste, large intervals could cause only "big" problems to be discussed and

"small" problems to be largely neglected. (6) When an existing development process is used in the background, the successful incorporation of agile principles requires the definition of similarities and connections to it, with phases merging into one another. (7) Particular attention must be paid to the whole environment as the introduction of agile approaches goes beyond a development team. There needs to be certainty that the information flow is sufficient over the whole supply chain. (8) It is often not clear who should be a member of the core agile team and who should attend what additional meetings. (9) Since rapid prototyping is a promising way to quickly transform concepts and ideas into physical products, it should be used more often.

Table 2. Challenges, advantages, and principles of agile product development.

No.	Challenges	Advantages	Principles
1	Human factor	Coping with incomplete and missing customer requirements	Internal exchange through interdisciplinary core teams and daily meetings
2	Many meetings with a lot of stakeholders	Creates transparency	External exchange through Product Manager
3	Customer and supplier Involvement	Enforced communication and feedback	Frequent delivery through short iterative development in numerous cycles
4	Integration in company partner's line organization	"One-Person Development" is prevented	Self-organized teams
5	Identification and integration of all involved departments	Individual team responsibility for decisions	Balance and control: agile development methodology with underlying development process
6	Integration with existing development process	Urge to deliver something worth showing after a sprint	Open-minded management support and commitment
7	Managing intersections of stakeholders and departments		Voluntary participation
8	Team constellation		
9	Rapid prototyping implementation		

3.2.2. Advantages

(1) Agile development helps with coping with changing and missing customer requirements. (2) Furthermore, it has turned out that the biggest advantage of agile development is not delivering the product earlier, but generating transparency. (3) Communication is strengthened by the numerous coordination meetings so that it is always clear who is currently doing what and who has completed which task. The increased feedback enables continuous process and product improvement. (4) By working in shorter development cycles, errors are detected earlier and faster. In the conventional development process, little coordination between the major milestones could cause certain developers to work in the wrong direction for a longer time. (5) It was always emphasized that teams have more decision-making autonomy but also responsibility, which enables them to make decisions faster and more consciously. (6) Since Scrum works with short cyclical sprints and an increment must be delivered after each sprint, the developers are required to create something to show from the beginning.

3.2.3. Principles

(1) It was recognized that a key way to promote internal communication is the formation of interdisciplinary core development teams. These teams meet daily to coordinate the work tasks according to the Scrum framework. (2) In addition to the internal exchange, information exchange with suppliers and customers through the product manager is essential. (3) After each sprint, a "presentable" result is to be submitted, which then flows

into the next development phase through generated learnings. The goal is to quickly get to the hardware development in an early stage by a forced need of solutions. (4) In the interviews, the self-organization and personal responsibility of the teams were always confirmed and emphasized. This means that project decisions have always been made by the group and not individuals, whereas decisions always have to be justified and the groups must take responsibility. (5) Another important principle is the balance of the agile and the underlying conventional development process. While the conventional process provides information on what has to be done and when, the agile framework directs the focus and method. (6) An open-minded organizational culture and commitment are sometimes named as the greatest enablers. This includes the provision of coaching opportunities as support. (7) Allowing every employee to volunteer for pilot projects strengthens motivation and does not impose a new development framework on any employee. In this context, challenge number one, the "human factor," should be mentioned.

3.3. Beta Division: Fundamentals of Agile Powertrain Development

Beta's main task is the development of automotive powertrain systems, whereas each project represents an individual and independent development service. Therefore, there is a substantial difference to Alpha division, as the development process itself is the product Beta offers. Apparently, Beta has not implemented any agile development principles and a conventional development process has been used for many years. In order to investigate the agile considerations, attributes and disciplines of powertrain development, semistructured interviews with one expert for design and quality, one development engineer, one systems engineer, one project manager, one powertrain integration engineer, one component engineer, and two project planners were conducted, each with an approximate duration of one hour.

3.3.1. Basics for Agile Powertrain Development

Considering agile powertrain development in Beta division, a critical challenge is the integration into the existing conventional process. The main benefit of agile powertrain development is not a shortened time-to-market, but rather improved fulfillment of customer needs as the SOP upon which the entire development strategy is set up will remain unchanged. As the existing development process only prescribes what has to be delivered and when, it represents a potential basis for agile development, focusing on the how. Therefore, the defined quality gates could remain in place, ensuring a controlled progress. In general, it is recognized that certain principles that can be described as agile are already in use but simply not labeled as such. First, feedback is requested from the customer after each milestone, even though this still occurs in larger intervals. Simultaneous engineers hold weekly meetings, whereas the customer and suppliers are involved occasionally. Small core teams, decoupled from the actual development process used for specific problems that suddenly arise, are similar to the teams employed in agile approaches. Here, these teams work over a short period of time in a highly iterative manner, with three to four development loops per week, using techniques such as rapid prototyping to solve specific problems. One of the current challenges in powertrain development is the coordination between software and hardware, since these are often out of synchronization. In general, agile development principles should not only focus on hardware but on an overall system combining both software and hardware. Rapid prototyping can only be used for the first generation of prototypes due to the fact that further generations require the use of serial tools. As hardware prototyping is, in general, extremely expensive and time-consuming, the aim is to perform as much of the front-loading work as possible through simulations. However, specific tests need to be done with hardware. While it is possible to enter the market with semifinished software, no compromises can be made when hardware is involved, especially due to safety aspects.

3.3.2. Attributes of Powertrain Development

In the next step, attributes that influence the development of powertrain systems are identified and presented in Table 3. These attributes derive from several factors such as:

- The vehicle targets that are given by the OEM.
- The development framework that is used by the customer, as the powertrain development process needs to be closely linked to the vehicle development process.
- The setting of system borders and the considerations of the division of responsibilities between the OEM, the client, suppliers, and all other stakeholders.
- The OEM's profile: e.g., the geographical origin.
- Interfaces to the vehicle.

Table 3. Attributes of powertrain development.

No.	Attribute
1	Vehicle Development (new, upgrade, derivate)
2	Powertrain Topology (ICE, battery, fuel cell, gearbox, etc.)
3	Degree of Maturity Entry (from scratch, after feasibility phase, into development phase)
4	Functional Requirements (weight, lifetime, power, consumption, etc.)
5	Legislative Requirements (emission standards, safety, etc.)
6	Generation/ Sample Quantity (2–6)
7	Hardware Test Quantity
8	Excluded Systems/Components
9	Testing (Virtual vs. Real)
10	Timeframe (24–48 months)

The first attribute is the vehicle development itself. It can either be a new vehicle, an upgrade or a derivate. Secondly, there is the powertrain topology, which can include different elements such as the ICE, the e-drive, a battery or gearbox, etc. Next, the degree of maturity where the project is entered can either be from scratch, after the feasibility phase, or right into the development phase. Furthermore, attributes that significantly influence the powertrain development are functional and legislative requirements, derived, for instance, from the vehicle targets and the customer profile. Moreover, attributes include the number of generations to be built, the hardware test quantity, systems and components that are excluded from the development, the choice between virtual and real testing, and the timeframe of the powertrain development.

3.3.3. Disciplines Involved in Powertrain Development

Identifying the disciplines required for the powertrain development is essential as interdisciplinary coordination is a challenge in agile approaches. An important role is played by project managers, who are responsible for communication with the customer, the understanding of customers' needs, and the development processes. Moreover, system engineers are mainly responsible for breaking down the requirements. The identified main disciplines for powertrain development are:

- Design;
- Service (Testing);
- Systems Engineering;
- Electric and Electronics Engineering;
- Mechanical Engineering;
- Functional Safety;
- Production;
- Software;

- Purchasing;
- Homologation.

"Simulation" as its own discipline was not taken into account as basically all departments have to carry out their own simulations.

3.4. Comparison of Development Principles in Alpha and Beta

As mentioned before, powertrain development characteristics identified in Beta partially correspond to agile principles. In order to understand this connection better, the fulfillment of the seven agile principles in Alpha was qualitatively assessed for Beta. This assessment was performed by the authors based on their insights built through the conducted interviews. Furthermore, the connections between these principles and the 12 agile manifesto principles were also defined by the authors. Figure 3 illustrates an overview of the results, and a detailed description can be found in the following paragraph.

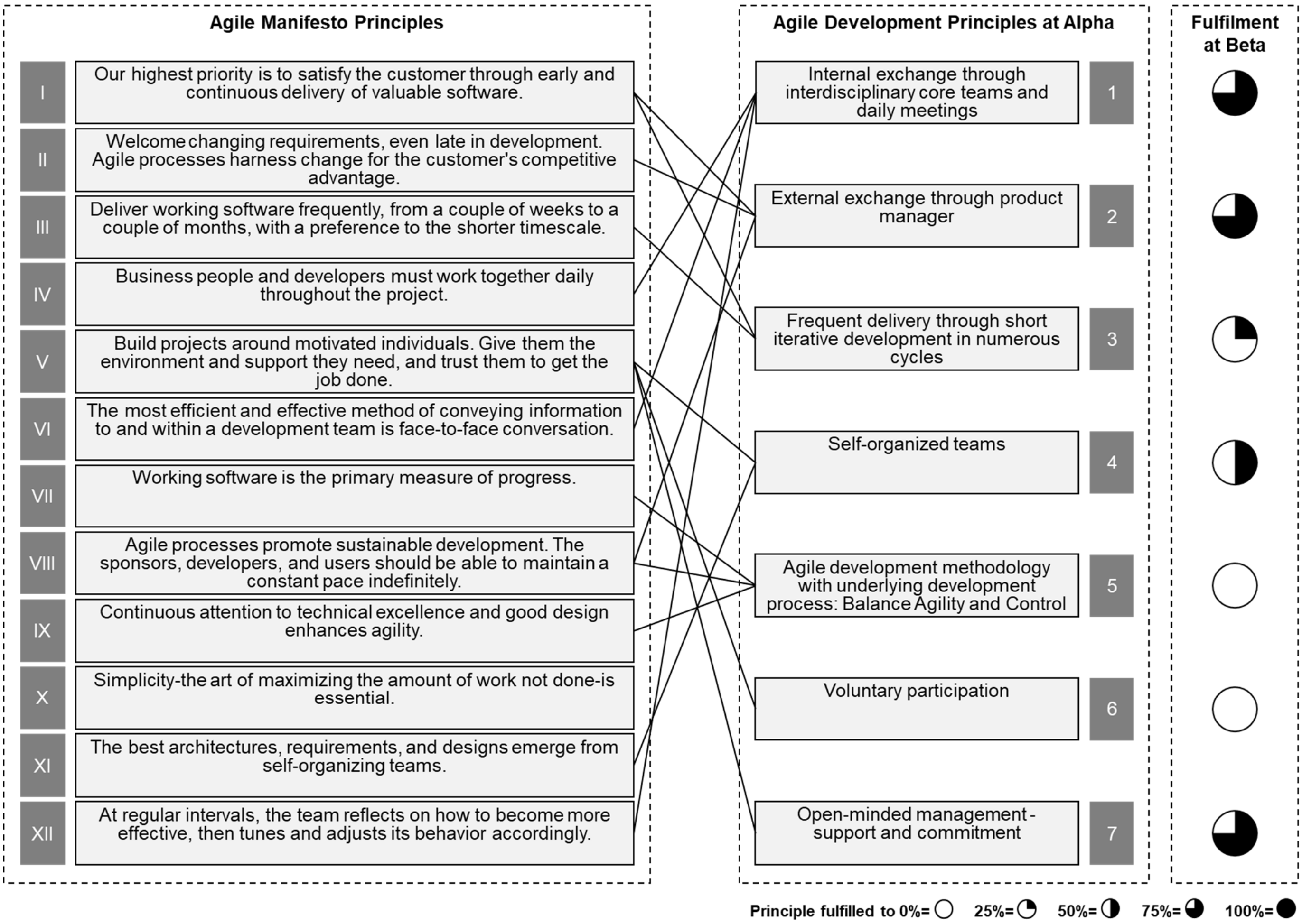
Agile Manifesto Principles
I
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
II
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
III
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
IV
Business people and developers must work together daily throughout the project.
V
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
VI
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
VII
Working software is the primary measure of progress.
VIII
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
IX
Continuous attention to technical excellence and good design enhances agility.
X
Simplicity-the art of maximizing the amount of work not done-is essential.
XI
The best architectures, requirements, and designs emerge from self-organizing teams.
XII
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Agile Development Principles at Alpha
Internal exchange through interdisciplinary core teams and daily meetings
1
External exchange through product manager
2
Frequent delivery through short iterative development in numerous cycles
3
Self-organized teams
4
Agile development methodology with underlying development process: Balance Agility and Control
5
Voluntary participation
6
Open-minded management - support and commitment
7
Fulfilment at Beta
Principle fulfilled to 0%=○ 25%=◔ 50%=◑ 75%=◕ 100%=●

Figure 3. Correlation between agile manifesto principles and agile development principles at Alpha and their fulfillment at Beta.

(1) Corresponds with the agile manifesto's principles IV, VI, and XII. Compared to Beta, this principle was rated as 75% fulfilled. Interdisciplinary core teams already exist to some extent and internal exchange is happening. However, these teams do not meet on a daily basis. (2) Corresponds with the agile manifesto's principles I, II, and VIII. It was evaluated to be also 75% fulfilled at Beta as there is a project manager who is responsible for external communication to the customer. Coordination meetings with customer and supplier occur after each milestone to keep the external information exchange high. However, these meetings do not happen on a regular basis. (3) Corresponds with the agile manifesto's

principles I and III. This principle was assessed to be fulfilled by 25% in Beta. The reason is that, although development takes place over several generations, these intervals tend to be relatively long. (4) Corresponds with the agile manifesto's principles V and XI. In Beta, this principle is fulfilled by 50% as development teams act partly organized by themselves. (5) Corresponds with the agile manifesto's principles VII, VIII, and IX. As Beta has not implemented an agile approach yet, this principle was assessed to be not fulfilled. (6) Corresponds with the agile manifesto's principle V. Also, because Beta has not implemented an agile approach yet, this principle was assessed to be not fulfilled. (7) Also corresponds best with V. The conducted interviews made it clear that the management is interested in the incorporation of agile development principles and sees advantages to powertrain development. Since agile development principles have not yet been applied, this principle was rated at 75%, since 100% would require immediate implementation.

4. Procedural Model for Assessment of Project Development Approach Balance

The adapted procedural model from [37] builds a basis to support the incorporation of agile principles in the powertrain development process. This model describes linking product context criteria (degree of novelty and the degree of interdependence) with project context criteria (characteristics of agile and conventional development) in a matrix. The aim is to provide an indication of whether a certain combination of attributes and project conditions is more suitable for the application of agile or conventional development principles and to support finding the right balance between agility and control.

First, the identified attributes of powertrain development (Table 3) and the required disciplines involved in powertrain development are listed in a domain mapping matrix (DMM-DA, with indices D for disciplines and A for attributes), described by [38]. Having defined the attributes of the specific powertrain development project, the subjective novelty value for the acting disciplines in a project must be assessed from discipline representatives themselves between 0, 1, and 2. The meanings of these numbers are as follows [37]:

- "2": Attribute has not yet been implemented; increased effort is to be expected with regards to implementation; feasibility cannot be assessed as clearly positive; very high communication effort to be assumed.
- "1": Attribute has not yet been implemented; no particular obstacles are to be expected in the implementation of the attribute; feasibility can be assessed as positive; increased communication effort.
- "0": Attribute (or similar) has already been implemented; simple implementation; low communication effort.

Table 4 shows an excerpt of this step of the procedural model. While selected sample attributes for this illustration are highlighted in green, sample subjective novelty values complete the DMM-DA. It is recommended that all involved lead engineers mutually select appropriate numbers for their related disciplines to actually reflect the commonly perceived novelty in practice. Votings or average calculations might lead to acceptable results in case of disagreements. Table 4 only represents an individual assessment to support the procedural understanding and therefore provides no general validity.

Table 4. Procedural model excerpt: domain mapping matrix—disciplines and attributes (sample values).

			Key Disciplines Involved in Powertrain Development									
			Design	**Service**	**Systems Engineering**	**Electrical Engineering**	**Mechanical Engineering**	**Functional Safety**	**Production**	**Software**	**Purchasing**	**Homologation**
Key Attributes of Powertrain Development	Vehicle Development	New	2	0	2	1	1	2	2	2	2	2
		Upgrade										
		Derivate										
	Powertrain Topology	ICE	2	2	1	2	2	2	2	2	2	2
		Battery										
		Fuel Cell										
		E-Motor										
		…										
	Degree of Maturity Entry	From scratch	1	2	0	2	1	2	1	2	0	2
		Feasibility										
		Development										
		…	…	…	…	…	…	…	…	…	…	…

Note: Novelty values (green).

After the successful assessment, a design structure matrix (DSM-D, with indices D for disciplines) is derived in the next step. The DSM-D maps the interrelation of the disciplines to each other and is calculated by multiplying the DMM-DA (Table 4) with its transposed version according to the following formula [39]:

$$\text{DSM-D} = \text{DMM-DA}^{T} \times \text{DMM-DA}. \qquad (1)$$

The resulting DSM-D shown in Table 5 provides two different types of information. First, the novelty values in the main diagonal of the matrix (highlighted in green) correspond to the subjective novelty value of each discipline. The higher the number, the more novel the project in question (summarizing all key attributes from Table 4) is in the specific discipline. Furthermore, the interdependency values in the lower triangular matrix (highlighted in orange) provide information about the correlation of the subjective novelty values for certain discipline pairs. Similar to before, higher values indicate more novelty.

The degree of novelty (DGN) represents the ratio of the novelty value for a specific discipline (e.g., service = 26) to the maximum possible novelty value of the matrix and is calculated according to Equation (2). While this maximum depends on the actual size of the DMM, in this example it is 40. As the realization of attributes with a higher subjective novelty value requires increased effort for the specific discipline, a high degree of novelty also indicates increased effort in general and the realization of these attributes in the powertrain development cannot take place without proper coordination within the related disciplines. The degree of interdependency (DGI) is used to evaluate the dependencies of correlated disciplines. Comparing the ratio of the interdependency value between two specific disciplines (e.g., service/design = 23) to the maximum possible value of the matrix (40), it is calculated according to Equation (3). According to Egelhoff (1991), it provides information about how independently a discipline can act for the realization of a certain attribute [40]. The DGI can therefore be used to reflect the need for coordination and communication between individual disciplines. Regarding the 10 identified key disciplines, a total of 45 combinations of DGIs can occur after the calculation.

$$\text{DGN [\%]} = \text{Novelty Value/Maximum Matrix Value} \tag{2}$$

$$\text{DGI [\%]} = \text{Interdependency Value/Maximum Matrix Value} \tag{3}$$

Table 6 shows the DSM-D with the calculated degrees of novelty and interdependency. A threshold value is further introduced to define a distinction between the recommendation of a conventional or agile approach. While a DGI above 25% means that agile principles are preferable for the proper realization of the product context, a DGI of 25% or lower points to conventional development principles. As the sample DGIs in Table 6 are all greater than 25%, only an agile approach is appropriate. The degrees of novelty are from this point onwards no longer considered, as the application of principles related to agility within a certain discipline is both easier and more common.

Table 5. Procedure model excerpt: domain structure matrix–disciplines.

		Key Disciplines									
		Design	**Service**	**Systems Engineering**	**Electrical Engineering**	**Mechanical Engineering**	**Functional Safety**	**Production**	**Software**	**Purchasing**	**Homologation**
Key Disciplines	Design	28									
	Service	23	26				Novelty Values				
	Systems Engineering	22	15	21			Interdependency Values				
	Electrical Engineering	27	25	22	31						
	Mechanical Engineering	22	20	18	25	23					
	Functional Safety	28	27	20	28	22	33				
	Production	31	26	26	32	25	32	37			
	Software	28	24	24	31	24	30	34	34		
	Purchasing	25	19	23	25	20	24	30	28	29	
	Homologation	29	26	23	31	25	32	34	33	26	34

Note: Novelty values (green) and interdependency values (orange).

Table 6. Procedure model excerpt: degrees of novelty and interdependency.

		Key Disciplines									
		Design	**Service**	**Systems Engineering**	**Electrical Engineering**	**Mechanical Engineering**	**Functional Safety**	**Production**	**Software**	**Purchasing**	**Homologation**
Key Disciplines	Design	70%									
	Service	58%	65%				Degrees of Novelty (DGN)				
	Systems Engineering	55%	38%	53%			Degrees of Interdependency (DGI)				
	Electrical Engineering	68%	63%	55%	78%						
	Mechanical Engineering	55%	50%	45%	63%	58%					
	Functional Safety	70%	68%	50%	70%	55%	83%				
	Production	78%	65%	65%	80%	63%	80%	93%			
	Software	70%	60%	60%	78%	60%	75%	85%	85%		
	Purchasing	63%	48%	58%	63%	50%	60%	75%	70%	73%	
	Homologation	73%	65%	58%	78%	63%	80%	85%	83%	65%	85%

Note: Novelty values (green) and interdependency values (orange).

Besides the product context criteria, the selection of an appropriate development approach is also subject to project- and environment-specific criteria. In order to evaluate whether project context criteria are more suitable for conventional or agile development principles, their differentiation characteristics (presented in Table 1) are used. Finally, the product and project context criteria are linked in a so-called Product–Project Context matrix (PPC), as shown in Figure 4. While the left side lists the differentiation criteria from Table 1 depending on their agile or conventional nature in the specific project, the bottom includes the agile or conventional discipline pairs according to Table 6. The product and project context should (ideally) coincide at an intersection in the gray fields of the shown matrix. If there is an intersection in a white field, this means the product and project–context do not properly correlate for a certain project approach. However, deriving recommended actions to move intersections from a white field to the gray one is not within the scope of this project.

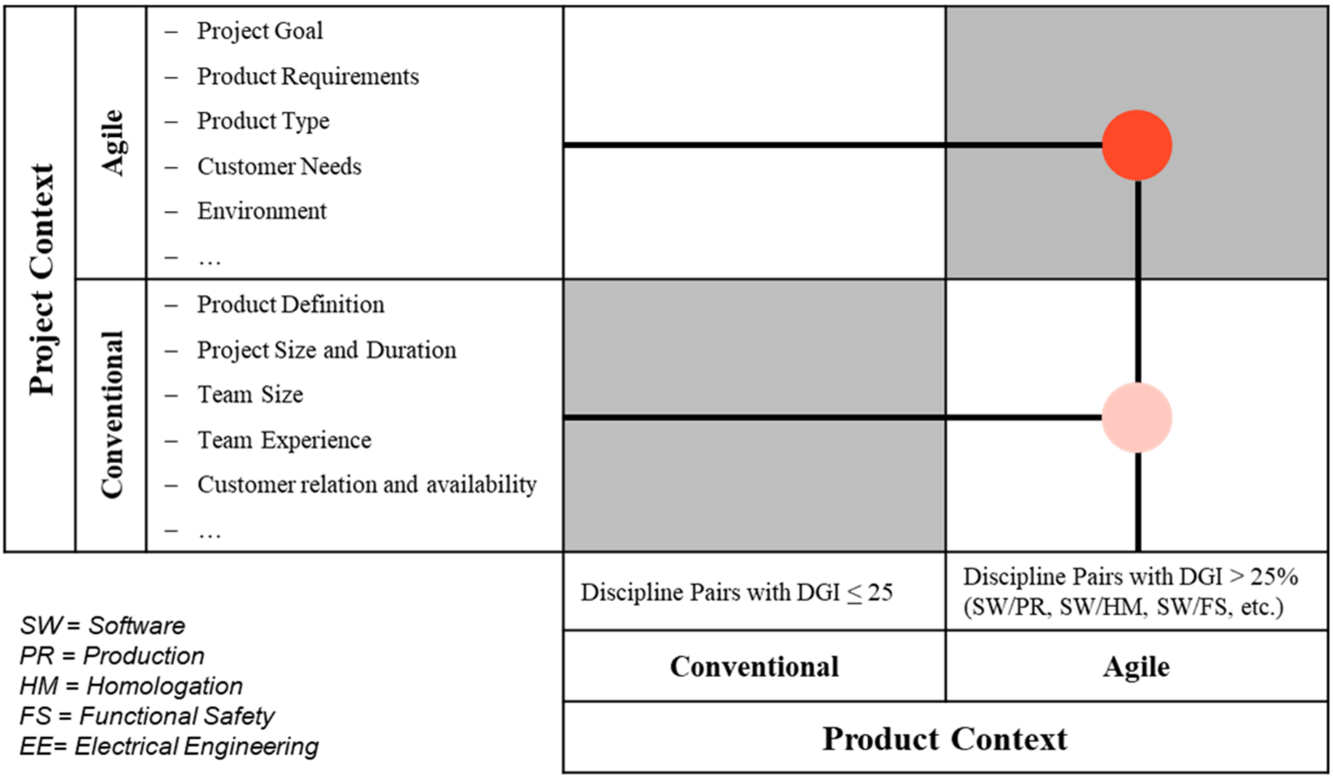

Figure 4. Product–Project Context.

5. Discussion

Having completed a development support tool, two reasons often hinder a full evaluation according to [41]. First, a lack of the required maturity of the support tool for its actual application in practice and second, a limiting research project duration, which was an obstacle in the present work. A full evaluation, including the application of the developed procedural model to a specific project, would have significantly exceeded the timeframe. Therefore, the actual evaluation was performed in two phases. The first phase included additional semistructured interviews, performed after completing each main interview with the experts at the investigated engineering company. This enabled us to gather valuable feedback, whereas its iterative implementation gradually improved the procedural model to satisfy the actual needs of future potential users. In the second phase, a semistructured interview with an experienced engineer was conducted as a final evaluation. The evaluation questions were:

- Is the procedural model applicable to the development of powertrain systems?

- Does the procedural model support the incorporation of agile development into an existing powertrain development process?
- Do you consider the disciplines/attributes as suitable?

The procedural model was considered as structured and systematic support for decisions. While some changes to the terminology would be required to fit the company's internal terminology exactly, its applicability to the development of powertrain systems was clear. In the opinion of the evaluation partner, the procedural model should not provide rigid recommendations for a certain project with either "conventional" or "agile" principles. However, the developed tool allows this desired flexibility and only provides a certain "target picture" to be followed. Referring to individual points in the project context, it presents deviations from the "ideal picture" of agile approaches in terms of the project environment. For example, the partner company would certainly want to carry out large projects with a high number of employees and developers in an agile manner, but an agile project execution is preferable for small projects and teams. For this purpose, agile teams could be divided according to the functional tasks required in a development project of a powertrain system. Nevertheless, the procedural model can still unveil fields that could cause complications, as their characteristics do not preferably fit into agile development. The development teams would know in advance where there is still a need for clear step-by-step development to find the right balance between control and agility.

6. Conclusions

This paper deals with the considerations of agile powertrain development as well as the incorporation of agile principles into conventional powertrain development processes. In contrast to the well-known and established agile development approaches in the software industry, agile development principles have only been partially established for hardware. However, no study was found that deals specifically with the highly relevant topic of agile powertrain development. Having investigated existing conventional product development approaches in the automotive industry, an extensive literature review was performed to define the characteristics for conventional and agile product development. This comparison has revealed the specific conditions when each approach is more likely to succeed. Furthermore, a single case study was conducted in close cooperation with an engineering company in the relevant business sector, whereas semistructured interviews served for data collection in a qualitative manner. For this study, two divisions called Alpha and Beta were investigated. As Alpha is already using an agile development approach, it enabled us to identify corresponding principles, advantages, and challenges. On the other hand, Beta deals with the development of powertrain systems and built the basis for the identification of related key disciplines. Additionally, certain powertrain attributes that have an explicit impact on development were defined. During the interviews it became clear that, throughout the conventional development of powertrain systems, certain characteristics or working methods are similar to agile principles. In order to better understand this connection, the fulfillment of the seven agile principles in Alpha was qualitatively assessed for Beta. Furthermore, the connections between these principles and the 12 agile manifesto principles were also defined. Finally, a procedural model from the literature was adapted to assess whether a certain combination of attributes and project conditions is more suitable for the application of an agile or conventional development approach. The procedural model supports balancing development flexibility (agile principles) and control (conventional principles). Regarding future pilot teams that might work with agile principles for powertrain development, it is supposed to illustrate a certain ideal picture and to create an understanding of whether agile or conventional development principles are more suitable for specific projects. Furthermore, it can point out problem areas in the project context that should be taken into account. The developed procedural model was evaluated in two phases, and its applicability for the development of powertrain systems was confirmed.

Author Contributions: All authors contributed to the study conception and design. A.L. and O.M.-T. were mainly responsible for data collection and analysis. L.S. and H.P.S. focused on the literature review. M.W. and C.R. provided valuable feedback on previous manuscript versions. All authors have read and agreed to the published version of the manuscript.

Funding: This work was conducted as part of the research project P2-Opti (Product- and production optimization covering the entire automotive powertrain lifecycle, 858656), which was funded by the Austrian Research Promotion Agency (FFG).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all interview participants involved in the study.

Data Availability Statement: A detailed overview of the relevant literature is provided. This theoretical perspective is complemented by practical insights through interviews conducted in two company divisions dealing with powertrain development and agile product development. The data are not available via any other repository.

Acknowledgments: This work was conducted as part of the research project P2-Opti (Product- and production optimization covering the entire automotive powertrain lifecycle). Sincere thanks to the Austrian Research Promotion Agency (FFG) for the project funding.

Conflicts of Interest: The authors state that this work and its results have not been published before and are not being submitted to any other journal. The authors declare they have no financial interests regarding this publication.

References

1. Buckley, P.; Verbeke, A.; Jankowska, B.; Van Tulder, R. *International Business in a VUCA World: The Changing Role of States and Firms*, 1st ed.; Progress in International Business Research; Emerald Publishing Limited: Bingley, UK, 2020.
2. Cooper, R.G. What's Next?: After Stage-Gate. *Res. Technol. Manag.* **2014**, *57*, 20–31. [CrossRef]
3. Smith, P.G. *Flexible Product Development: Building Agility for Changing Markets*; John Wiley & Sons/Jossey-Bass: San Francisco, CA, USA, 2007.
4. Ramsauer, C.; Kayser, D.; Schmitz, C. Erfolgsfaktor Agilität. In *Chancen für Unternehmen in Einem Volatilen Marktumfeld*, 1st ed.; Wiley-VCH: Weinheim, Germany, 2017.
5. Schuh, G.; Diels, F.; Ortlieb, C.; Riesener, M.; Schröder, S. Agile Produktentwicklung. In *Internet of Production für Agile Unternehmen*; Brecher, C., Klocke, F., Schmitt, R., Schuh, G., Eds.; AWK Aachener Werkzeugmaschinen-Kolloquium; Apprimus Verlag: Aachen, Germany, 2017; pp. 27–52.
6. Žužek, T.; Gosar, Z.; Kušar, J.; Berlec, T. Adopting Agile Project Management Practices in Non-Software SMEs: A Case Study of a Slovenian Medium-Sized Manufacturing Company. *Sustainability* **2020**, *12*, 9245. [CrossRef]
7. Volkswagen: Volkswagen Konzern Setzt auf Neue Formen der Zusammenarbeit. Hg. v. Volkswagen, 2017. Available online: https://www.volkswagenag.com/de/news/2017/07/Volkswagen_Konzern_setzt_auf_neue_Formen_der_Zusammenarbeit.html (accessed on 5 June 2020).
8. Lührig, T. Risikomanagement in der Produktentwicklung der Deutschen Automobilindustrie: Von der Konzeptentwicklung bis zum Produktionsanlauf. Ph.D. Thesis, Technical University Darmstadt, Darmstadt, Germany, 2006.
9. Schömann, S.O. *Produktentwicklung in der Automobilindustrie: Managementkonzepte vor dem Hintergrund Gewandelter Herausforderungen*; Gabler Verlag/Springer Fachmedien Wiesbaden GmbH (Gabler Research): Berlin, Germany, 2011.
10. VDI 2221-Part 1. *Design of Technical Products and Systems*; Verein Deutscher Ingenieure: Düsseldorf, Germany, 2019.
11. Cooper, R.G.; Sommer, A.F. Agile-Stage-Gate: New idea-to-launch method for manufactured new products is faster, more responsive. *Ind. Mark. Manag.* **2016**, *59*, 167–180. [CrossRef]
12. Cooper, R.G. Stage-gate systems: A new tool for managing new products. *Bus. Horiz.* **1990**, *33*, 44–54. [CrossRef]
13. VDI 2206. *Design Methodology for Mechatronic Systems*; Verein Deutscher Ingenieure: Düsseldorf, Germany, 2004.
14. Boehm, B.; Turner, R. Balancing agility and discipline. In *A Guide for the Perplexed*; Addison-Wesley: Boston, MA, USA, 2004.
15. Conforto, E.C.; Amaral, D. Agile project management and stage-gate model—A hybridframework for technology-based companies. *J. Eng. Technol. Manag.* **2016**, *40*, 1–14. [CrossRef]
16. Conforto, E.C.; Amaral, D.C.; da Silva, S.L.; Di Felippo, A.; Kamikawachi, D.S.L. The agility construct on project management theory. *Int. J. Proj. Manag.* **2016**, *34*, 660–674. [CrossRef]
17. Schuh, G. Agile Product Development. In *International Benchmarking Study*; Fraunhofer IPT, RWTH Aachen University: Aachen, Germany, 2019.
18. Kantelberg, J. Gestaltung Agiler Entwicklungsprozesse Technischer Produkte. Design of Agile Product Development Processes. Ph.D. Thesis, RWTH Aachen University, Aachen, Germany, 2018.

19. Takeuchi, H.; Nonaka, I. The new new product development game. Stop running the relay race and take up rugby. *Harv. Bus. Rev.* **1986**, *64*, 137–146.
20. Klein, T. *Agiles Engineering im Maschinen—Und Anlagenbau*; Herbert Utz Verlag (Forschungsberichte IWB, 323): München, Germany, 2016.
21. Cobb, C.G. Making Sense of Agile Project Management. In *Balancing Control and Agility*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2011.
22. Fernandez, D.J.; Fernandez, J.D. Agile Projectmanagement—Agilism Versus Traditional Approaches. *J. Comput. Inf. Syst.* **2008**, *49*, 10–17. [CrossRef]
23. Wysocki, R.K. Effective project management. In *Traditional, Agile, Extreme*, 7th ed.; John Wiley & Sons: Indianapolis, IN, USA, 2014.
24. Habermann, F. Hybrides Projektmanagement—Agile und klassische Vorgehensmodelle im Zusammenspiel. *HMD* **2013**, *50*, 93–102. [CrossRef]
25. Hartel, D.H. *Projektmanagement in Logistik und Supply Chain Management*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2019.
26. Nerur, S.; Balijepally, V.G. Theoretical reflections on agile development methodologies. *Commun. ACM* **2007**, *50*, 79–83. [CrossRef]
27. Schmidt, T.S.; Paetzold, K. Agilität als Alternative zu traditionellen Standards in der Entwicklung physischer Produkte: Chancen und Herausforderungen. Design for X—Beiträge zum 27. *DfX-Symp. Oktober* **2016**, 255–267.
28. Špundak, M. Mixed Agile/Traditional Project Management Methodology—Reality or Illusion? *Procedia—Soc. Behav. Sci.* **2014**, *119*, 939–948. [CrossRef]
29. Book, M.; Gruhn, V.; Striemer, R. *Erfolgreiche Agile Projekte*; Springer: Berlin/Heidelberg, Germany, 2017.
30. Brehm, L.; Feldmüller, D.; Rieke, T. Konfiguration des hybriden Projektmanagements für die Entwicklung technischer, physischer Produkte. In *Angewandte Forschung in der Wirtschaftsinformatik Prozesse, Technologie, Anwendungen, Systeme und Management*; Barton, T., Herrmann, F., Meister, V., Müller, C., Seel, C., Eds.; Mana-Buch: Heide, Germany, 2017; pp. 30–39.
31. Welge, M.; Friedrich, C.; Shair, A. Integration von agilen Methoden in der Systementwicklung. In *Tag des Systems Engineering*; Maik, M., Schulze (Hg.), S.-O., Eds.; Zusammenhänge Erkennen und Gestalten; Paderborn 7–9 November 2012; Hanser (Hanser eLibrary): München, Germany, 2013; pp. 341–350.
32. Nerur, S.; Mahapatra, R.; Mangalaraj, G. Challenges of migrating to agile methodologies. *Commun. ACM* **2005**, *48*, 72–78. [CrossRef]
33. Edmondson, A.C.; McManus, S.E. Methodological fit in management field research. *Acad. Manag. Rev.* **2007**, *32*, 1155–1179. [CrossRef]
34. Ahlström, P. Research methods for operations management. In *The Research Process*; Karlsson, C., Ed.; Taylor & Francis: New York, NY, USA, 2016; pp. 46–78.
35. Voss, C.; Johnson, M.; Godsell, J. Research Methods for Operations Management. In *Case Research*; Karlsson, C., Ed.; Taylor & Francis: New York, NY, USA, 2016; pp. 165–194.
36. Yin, R.K. *Case Study Research: Design and Methods*, 4th ed.; Sage: Thousand Oaks, CA, USA, 2009.
37. Schnöll, H.P. Integrierte Produktentstehung: Ein Vorgehensmodell zur Kontextspezifischen Gestaltung des Produktentstehungsprozesses von Bauteilen aus Faserverbundkunststoffen. Ph.D. Thesis, Graz University of Technology, Graz, Austria, 2015.
38. Eppinger, S.D.; Browning, T.R. *Design Structure Matrix Methods and Applications*, 1st ed.; The MIT Press: Cambridge, MA, USA, 2012.
39. Lindemann, U.; Maurer, M.; Braun, T. *Structural Complexity Management: An Approach for the Field of Product Design*, 1st ed.; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
40. Egelhoff, W.G. Information-Processing Theory and the Multinational Enterprise. *J. Int. Bus. Stud.* **1991**, *22*, 341–368. [CrossRef]
41. Blessing, L.T.M.; Chakrabarti, A. *DRM, a Design Research Methodology*; Springer: London, UK, 2009.